



**SISTEM PROTEKSI DAYA LISTRIK BERBASIS MIKROKONTROLER
ATMEGA 16 DENGAN SENSOR ARUS**

PROYEK AKHIR

**Diajukan Kepada Dekan Fakultas Teknik Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Ahli Madya Teknik**



Disusun Oleh:

PUTU DARSANA

10507131024

**PROGRAM STUDI TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

2015

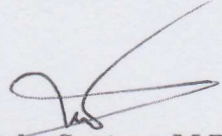
HALAMAN PERSETUJUAN

Proyek akhir yang berjudul “Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus” ini telah disetujui oleh pembimbing untuk diujikan



Menyetujui

Pembimbing



Drs. Djoko Santoso, M.Pd
NIP. 19580422 198403 1 002

LEMBAR PENGESAHAN

Proyek akhir yang berjudul “Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus” ini telah dipertahankan di depan dewan penguji pada tanggal ~~11~~ 17-Februari - 2015 dan dinyatakan lulus.

DEWAN PENGUJI			
Nama	Jabatan	Tanda Tangan	Tanggal
Drs. Djoko Santoso, M.Pd	Ketua Penguji		17/2 2015
Satriyo Agung Dewanto, M.Pd	Sekretaris Penguji		17/2 2015
Dr. Putu Sudira, M.P	Penguji Utama		17/2 2015

Yogyakarta, Februari 2015
Fakultas Teknik Universitas Negeri
Yogyakarta

Dekan,

Dr. Moch. Bruri Triyono, M. Pd.
NIP. 19560216 198603 1 003

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini

Nama : Putu Darsana

NIM : 10507131024

Program Studi : Teknik Elektronika

Judul PA : Sistem Proteksi Daya Listrik Berbasis Mikrokontroler
ATmega 16 dengan Sensor Arus

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis orang lain sebagai pesyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau Perguruan Tinggi lain, kecuali bagian – bagian tertentu yang saya ambil sebagai acuan dengan mengikuti tata cara dan penulisan karya ilmiah yang lazim. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, Februari - 2015

Penulis,



Putu Darsana

Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus

Oleh: PUTU DARSANA

Nim: 10507131024

ABSTRAK

Pembuatan proyek akhir ini bertujuan untuk membuat perangkat keras (*hardware*), perangkat lunak (*software*) dan untuk mengetahui unjuk kerja dari Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus.

Metode perancangan Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus ini menggunakan metode rancang bangun yang terdiri dari : (1) Identifikasi kebutuhan, (2) Analisis kebutuhan, (3) Perancangan Alat, (4) Pembuatan alat, dan (5) Pengujian. Pada proyek akhir ini, Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus dibuat sebagai alat proteksi yang berbasis mikrokontroler ATmega 16. Selain itu alat ini juga sebagai penampil nilai dari tegangan jala-jala PLN (Volt) , arus (Ampere) yang mengalir pada beban, nilai faktor daya (*cos phi*) sehingga didapatkan juga nilai daya nyata (Watt) yang akan ditampilkan pada LCD 16x2. Sistem proteksi pada alat ini juga dapat diatur nilai batas arus proteksinya dengan menggunakan *push button* pengaturan, dimana jika nilai batas arus yang ditentukan terlewati maka *relay* akan aktif untuk memutus arus kebeban dan *buzzer* berbunyi sebagai alarm.

Sistem ini terdiri atas perangkat keras dan perangkat lunak. Untuk perangkat keras terdiri dari : (1) *Power supply* simetris ± 8 V, catu daya tunggal +5 V, dan +12V (2) rangkaian sensor tegangan menggunakan trafo *step down* (3) rangkaian penguat instrumentasi dan sensor arus *Split Current Transformer CT301* (4) rangkaian pendeteksi beda fasa, (5) rangkaian sistem minimum mikrokontroler ATmega 16, (6) rangkaian *driver relay* dan *buzzer*, (7) *push button*, (8) serta penampil LCD 16x2. Untuk perangkat lunak dirancang sesuai rancangan perangkat keras yang terdiri dari program pembaca ADC dari sensor tegangan, ADC dari sensor arus, program interrupt untuk pembacaan nilai beda fasa, program perhitungan untuk mendapat nilai daya nyata, program penampil LCD 16 x 2, program *timer*, program menu menggunakan *push button*, serta program pengolah data untuk sistem proteksi. Kesimpulan yang didapat dari alat ini adalah alat sudah dapat bekerja namun masih memiliki beberapa kekurangan diantaranya batas ukur arus maksimal sebesar 2 A dan kurang akuratnya pembacaan nilai beda fasa yang disebabkan oleh perbedaan bahan dari sensor tegangan yang menggunakan inti besi sedangkan bahan dari sensor arus yang menggunakan batang ferit.

Kata kunci : Alat proteksi, Sensor Arus, Sensor Tegangan, Atmega 16, *cosphi*, Daya

MOTTO DAN PERSEMBAHAN

MOTTO

“Belajar itu butuh waktu dan proses walaupun berat jika dikerjakan terus – menerus akan ada hasilnya dari pada menganggap mudah jika tidak dikerjakan tidak akan ada hasilnya kerjakan kerjakan dan kerjakanlah”

(Penulis)

Dalam usaha ini tidak ada kerugian ataupun pengurangan, dan sedikitpun kemajuan dalam menempuh jalan ini dapat melindungi seseorang terhadap rasa takut paling berbahaya.

(Bhagawad Gita II.40)

PERSEMBAHAN

Proyek akhir ini saya persembahkan kepada :

Bapak, Ibu, adik dan Seluruh keluarga besar atas doa dan dukungan yang sangat memotivasi.

Teman – teman Kelas B 2010 Teknik Elektronika FT UNY. Terimakasih atas dukungan, bantuan, dan motivasinya dalam penyelesaian Proyek Akhir ini.

Keluarga Putra Bali (KPB) Purantara Yogyakarta

Sahabat KPB Pura Yogyakarta yang sering memberi motivasi dan mendoakan untuk segera menyelesaikan Proyek akhir ini.

KMHD UNY

Sahabat Keluarga Mahasiswa Hindu Dharma UNY, dan Pembina KMHD UNY Bpk Putu Sudira terimakasih atas doa dan motivasinya.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadiran Ida Sang Hyang Widhi Wasa / Tuhan Yang Maha Esa atas segala asung kerta wara nugraha beliau sehingga penulis dapat menyelesaikan penyusunan laporan Proyek Akhir dengan judul “Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus “. Pembuatan Proyek Akhir ini merupakan salah satu persyaratan memperoleh gelar Ahli Madya Program Studi Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta

Penulis menyadari bahwa dalam pembuatan Proyek Akhir ini dapat terselesaikan tak lepas dari berbagai pihak baik itu secara maupun secara tidak langsung . Untuk itu penulis mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Dr. Moch. Bruri Triyono, selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
2. Bapak Drs. Muhammad Munir, M.Pd, selaku Ketua Jurusan Pendidikan Teknik Elektronika.
3. Drs.Djoko Santoso, M.Pd, selaku Dosen Pembimbing Proyek Akhir.
4. Kedua Orang Tua yang telah memberi kesempatan untuk mencapai cita – cita.
5. Teman-teman seangkatan dan seperjuangan, terimakasih atas kerja samanya serta dukungannya.
6. Para Dosen, Teknisi Lab, dan Staff Jurusan Pendidikan Teknik Elektronika dan Informatika yang telah memberikan bantuannya.
7. Semua pihak yang telah membantu Proyek Akhir ini yang penulis tidak bisa sebut satu-persatu.

Penulis menyadari sepenuhnya bahwa dalam pembuatan Proyek Akhir ini masih banyak kekurangannya. Oleh karena itu kritik dan saran yang bersifat membangun untuk perbaikan Proyek Akhir ini senantiasa diterima dengan senang hati. Semoga laporan ini bermanfaat bagi siapa saja yang membacanya.

Yogyakarta, Februari -2015

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN	iv
ABSTRAK	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR TABEL	xvii
DAFTAR GAMBAR	xix
DAFTAR LAMPIRAN	xxii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	3
C. Batasan Masalah	3
D. Rumusan Masalah	4
E. Tujuan	4
F. Manfaat	5
G. Keaslian Gagasan	6

BAB II PENDEKATAN PEMECAHAN MASALAH	7
A. Sistem Proteksi Daya Listrik.....	7
B. Sensor	8
1. Linearitas	8
2. Tidak tergantung temperatur.....	9
3. Kepekaan	9
4. Waktu tanggapan	9
5. Batas frekuensi terendah dan tertinggi	9
6. Stabilitas waktu	9
7. Histerisis	10
C. Sensor Arus <i>Split Current Transformer CT301</i>	10
D. Mikrokontroler AVR (<i>Alf and Vegaard's Risc Processor</i>)	12
E. Transistor	17
1. Jenis Transistor	20
2. BJT.....	21
F. <i>Relay</i>	21
1. <i>Relay SPST (Single Pole Single Through)</i>	22
2. <i>Relay DPST (Double Pole Single Through)</i>	22
3. <i>Relay SPDT (Single Pole Double Through)</i>	22
4. <i>Relay DPDT (Double Pole Double Through)</i>	23
G. <i>Transformator (Trafo)</i>	23
H. <i>IC Regulator</i>	25
I. <i>Buzzer</i>	26

J. Gerbang XOR.....	27
K. Penguat Operasional (<i>OP – Amp</i>)	28
L. LCD <i>Display 16x2 M162</i>	29
M. Perangkat Lunak.....	31
1. Bahasa C.....	31
(a) <i>Header</i>	32
(b) Tipe Data	32
(c) Konstanta	33
(d) Label, Variabel, Fungsi	33
(e) Komentar	34
(f) <i>Reserved Keywords</i>	34
(g) Operator	34
(h) Aritmatika	35
(i) Logika	36
(j) Manipulasi Bit.....	36
(k) Percabangan	37
(1) <i>If – then</i>	37
(2) <i>If – then – else</i>	37
(3) <i>Switch – case</i>	37
(4) <i>Switch – case – default</i>	37
(5) Perulangan <i>for</i>	38
(6) <i>While</i>	38
(7) <i>Do – while</i>	39

(l) Konversi pola (%)	39
(m)Prosedur	40
(n) Fungsi.....	40
(o) Memasukkan Bahasa <i>Assembly</i>	40
2. <i>Code Vision AVR</i>	41
BAB III KONSEP RANCANGAN	43
A. Identifikasi Kebutuhan	43
B. Analisis Kebutuhan	44
C. Perancangan Sistem	44
1. Perancangan perangkat keras (<i>Hardware</i>).....	44
(a) Rangkaian Catu Daya Simetris	46
(b) Rangkaian Sensor Tegangan	47
(c) Rangkaian Penguat Instrumentasi	49
(d) Rangkaian Detektor Beda Fasa	50
(e) Rangkaian Sismin Mikrokontroler ATmega 16.....	52
(f) Unit Penampil (LCD text 16x2).....	53
(g) <i>Push Button</i>	54
(h) Rangkaian <i>Relay</i> dan <i>Buzzer</i>	55
2. Perancangan perangkat lunak (<i>Software</i>)	56
(a) Algoritma	57
(b) <i>Flowchart</i> program.....	60
D. Pembuatan Alat	61
1. Pembuatan <i>Printed Circuit Board (PCB)</i>	63

2. Pelarutan dan pengeboran PCB.....	64
3. Pengecekan jalur rangkaian.....	64
4. Pemasangan Komponen	64
5. Pembuatan <i>Box</i> rangkaian	65
E. Spesifikasi Alat	66
F. Pengujian Alat.....	66
1. Uji fungsionalitas	66
2. Uji keseluruhan system	67
G. Pengoperasian Alat.....	67
H. Jadwal Pelaksanaan.....	68
I. Daftar Komponen dan Biaya Pembuatan.....	69
 BAB IV PROSES,HASIL, PENGUJIAN, DAN PEMBAHASAN	70
A. Proses Pembuatan	70
1. Realisasi Program	70
2. Realisasi <i>Hardware</i>	70
a) Membuat desain rangkaian dan jalur PCB.....	70
b) Proses penyablonan pada PCB.....	72
c) Proses pelarutan	73
d) Proses pengeboran.....	74
e) Proses pemasangan komponen dan penyolderan	75
f) Proses perakitan rangkaian pada Box	75
B. Hasil	76
1. Hasil perakitan alat.....	76

a) Rangkaian Catu Daya.....	76
b) Rangkaian Sensor Tegangan.....	76
c) Rangkaian Penguat Instrummentasi.....	77
d) Rangkaian Pendeteksi Beda Fasa.....	77
e) Rangkaian Sistem Minimum ATmega 16.....	78
f) Rangkaian Driver Relay dan Buzzer.....	78
g) Hasil keseluruhan Alat	79
C. Pengujian	79
1. Pengujian Setiap Bagian	79
a) Pengujian Catu daya.....	79
b) Pengujian rangkaian sensor tegangan	80
c) Pengujian Sensor <i>Split Current Transformer</i>	81
d) Pengujian rangkaian Penguat Instrumentasi	82
e) Pengujian rangkaian Pendeteksi Beda Fasa	83
f) Pengujian rangkaian <i>Driver relay</i> dan <i>Buzzer</i>	85
g) Pengujian tampilan LCD.....	86
h) Pengujian <i>Push Button</i>	87
2. Pengujian keseluruhan	88
a) Pengujian Beban Resistif	88
b) Pengujian Beban Capasistif	89
c) Pengujian Beban Induktif.....	90
D. Pembahasan.....	91
1. <i>Hardware</i>	91

a)	Rangkaian Catu Daya.....	91
b)	Rangkaian Sensor Tegangan.....	91
c)	Sensor Arus <i>Split Current Transformer CT301</i> dan Rangkaian Penguat Instrumentasi	92
d)	Rangkaian Pendeteksi Beda Fasa.....	93
(1)	Rangkaian penguat Op-Amp sinyal tegangan..	93
(2)	Rangkaian penguat Op-Amp sinyal arus.....	94
(3)	Rangkaian gerbang logika.....	96
e)	Sistem Minimum Mikrokontroler ATmega 16	100
(1)	ADC (<i>analog to digital converter</i>).....	100
(2)	Interrupt	100
(3)	Push Button	101
(4)	Tampilan LCD	102
f)	Rangkaian <i>Driver Relay</i> dan <i>Buzzer</i>	103
2.	<i>Software</i>	103
a)	Definisi Prosesor	103
b)	Penyertaan Fungsi	103
c)	Definisi Variabel	104
d)	Definisi Mode Timer1	104
e)	Definisi Port dan Mode ADC.....	105
f)	Definisi Port dan Mode Interrupt	106
g)	Definisi Sub Rutin.....	107
(1)	Sub Rutin Push Button	107

(2) Sub Rutin Display	107
(3) Sub Rutin Pengolahan sampling beban.....	109
(4) Sub Rutin konversi Data	110
(5) Sub Rutin data Prosesing	110
 BAB V KESIMPULAN DAN SARAN	112
A. Kesimpulan	112
B. Keterbatasan alat	113
C. Saran	113
 DAFTAR PUSTAKA	115
LAMPIRAN.....	117

DAFTAR TABEL

Tabel 1. Fungsi-fungsi khusus Port B pada mikrokontroler ATmega16	15
Tabel 2. Fungsi-fungsi khusus Port D pada mikrokontroler ATmega16	17
Tabel 3. Konfigurasi PIN LCD 16x2	30
Tabel 4. Berbagai macam tipe data dalam <i>compiler Code Vision AVR</i>	32
Tabel 5. Simbol dan Aritmatika	35
Tabel 6. Simbol dan Pembanding	36
Tabel 7. Manipulasi Bit	36
Tabel 8. Jadwal pelaksanaan	68
Tabel 9. Daftar Komponen	69
Tabel 10. Hasil Pengujian Rangkaian Catu Daya	80
Tabel 11. Hasil Pengujian Rangkaian Sensor Tegangan	81
Tabel 12. Hasil perbandingan pengukuran tegangan antara multimeter digital dengan Sistem Proteksi Daya Listrik	81
Tabel 13. Hasil pengujian Sensor Arus <i>Split Current Transformer</i>	82
Tabel 14. Hasil Pengujian blok rangkaian Penguat Instrumentasi	82
Tabel 15. Hasil pengujian Rangkaian Pendeteksi Beda Fasa	83
Tabel 16. Hasil pengujian rangkaian <i>Driver Relay</i> dan <i>Buzzer</i>	85
Tabel 17. Pengujian beban Resistif dengan Lampu Bohlam	89
Tabel 18. Pengujian beban Kapasitif	89
Tabel 19. Pengujian menggunakan beban Induktif	90
Tabel 20. Keluaran rangkaian gerbang logika pada saat beban Resistif	97
Tabel 21. Keluaran rangkaian gerbang logika pada saat beban Induktif	98

Tabel 22. Keluaran rangkaian gerbang logika pada saat beban Capasistif	99
Tabel 23. <i>Port I/O</i> untuk operasi fungsi <i>Push Button</i>	102
Tabel 24. Konfigurasi <i>Pin</i> LCD 16 x 2	102

DAFTAR GAMBAR

Gambar 1. Bentuk fisik Sensor Arus <i>Split Current Transforrmer CT301</i>	11
Gambar 2. Spesifikasi Sensor <i>Split Current Transformer CT301</i>	12
Gambar 3. Konfigurasi PIN ATmega16	14
Gambar 4. Bentuk fisik Transistor	18
Gambar 5. Simbol Transistor	20
Gambar 6. Simbol <i>relay</i>	22
Gambar 7. Bentuk fisik Trafo	23
Gambar 8. Hubungan kumparan primer dan sekunder trafo.....	24
Gambar 9. Susunan kaki <i>IC Regulator</i>	25
Gambar 10. Rangkaian <i>power supply dengan IC regulator</i>	25
Gambar 11. Bentuk fisik <i>Buzzer</i>	27
Gambar 12. Gerbang XOR 2 Input	27
Gambar 13. Konfigurasi pin IC LM 741.....	28
Gambar 14. Rangkaian penguat <i>inverting</i>	29
Gambar 15. Bentuk fisik LCD M1632.....	31
Gambar 16. Blok diagram perancangan alat	45
Gambar 17. Rangkaian <i>Power Supply Unit</i>	47
Gambar 18. Rangkaian Sensor Tegangan	48
Gambar 19. Rangkaian penguat instrumentasi sensor arus.....	50
Gambar 20. Rangkaian penguat instrumentasi beda fasa.....	51
Gambar 21. Rangkaian komparator beda fasa arus.....	51

Gambar 22. Rangkaian penguat komparator beda fasa tegangan	52
Gambar 23. Rangkaian logika pemroses sinyal beda fasa tegangan terhadap arus	52
Gambar 24. Rangkaian Sistem Minimum Mikrokontroler ATmega 16	53
Gambar 25. Konfigurasi kaki LCD pada PIN ATmega 16.....	54
Gambar 26. Rangkaian <i>Push Button</i>	55
Gambar 27. <i>Driver output</i> motor dan <i>Buzzer</i>	56
Gambar 28. Gambar <i>flowchart</i>	60
Gambar 29. Desain <i>Box</i>	65
Gambar 30. Ilustrasi pembuatan rangkaian pada ISIS	71
Gambar 31. Ilustrasi pembuatan PCB pada ARES	72
Gambar 32. Proses penyablonan menggunakan setrika pada papan PCB	73
Gambar 33. Proses pelarutan PCB dengan larutan feri clorida.....	74
Gambar 34. Proses pengeboran PCB	74
Gambar 35. Hasil Rangkaian Catu Daya	76
Gambar 36. Hasil Rangkaian Sensor Tegangan.....	76
Gambar 37. Hasil Rangkaian Penguat Instrumentasi.....	77
Gambar 38. Hasil Rangkaian Pedeteksi Beda Fasa	77
Gambar 39. Hasil Rangkaian Sistem Minimum ATmega 16	78
Gambar 40. Hasil Rangkaian Driver Relay dan Buzzer	78
Gambar 41. Hasil Rangkaian Keseluruhan Alat	79
Gambar 42. Tampilan LCD pada simulasi.....	87
Gambar 43. Rangkaian komparator sinyal tegangan	93

Gambar 44. Rangkaian penguat <i>Op-Amp</i> sinyal arus	94
Gambar 45. Rangkaian gerbang Logika.....	96
Gambar 46. <i>Output</i> sinyal beban Resistif	97
Gambar 47. <i>Output</i> sinyal beban Induktif.....	98
Gambar 48. <i>Output</i> sinyal beban Kapasitif.....	99

DAFTAR LAMPIRAN

LAMPIRAN 1 Skema Rangkaian <i>Power Supply</i>	17
LAMPIRAN 2 Skema Rangkaian Sensor Tegangan	18
LAMPIRAN 3 Skema Rangkaian Penguat Instrumentasi	19
LAMPIRAN 4 Skema Rangkaian Pendeteksi Beda Fasa	20
LAMPIRAN 5 Skema Rangkaian Sistem Minimum Mikrokontroler ATmega16 ...	21
LAMPIRAN 6 Skema Rangkaian <i>Driver relay</i> dan <i>Buzzer</i>	22
LAMPIRAN 7 Tata letak komponen dan <i>layout</i> PCB Rangkaian <i>Power Supply</i>	23
LAMPIRAN 8 Tata letak komponen dan <i>layout</i> PCB	
Rangkaian Penguat Instrumentasi	24
LAMPIRAN 9 Tata letak komponen dan <i>layout</i> PCB	
Rangkaian Sistem Minimum Mikrokontroler ATmega 16	25
LAMPIRAN 10 Tata letak komponen dan <i>layout</i> PCB	
Rangkaian <i>Driver relay</i> dan <i>Buzzer</i>	25
LAMPIRAN 11 Tata letak komponen dan <i>layout</i> PCB	
Rangkaian Sensor Tegangan	26
LAMPIRAN 12 Tata letak komponen dan <i>layout</i> PCB	
Rangkaian Pendeteksi Beda Fasa	27
LAMPIRAN 13 <i>Listing</i> program lengkap.....	28

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Alat proteksi adalah suatu alat yang berfungsi untuk melindungi peralatan dari gangguan-gangguan sehingga bisa menghindari atau mengurangi kerusakan peralatan-peralatan listrik dan kejadian – kejadian yang tidak di inginkan misalnya: kebakaran. Semakin cepat reaksi suatu alat proteksi yang terpasang pada sebuah sistem maka akan semakin sedikit pengaruh gangguan terhadap alat atau sistem yang dilindungi dari kerusakan.

Alat proteksi listrik yang pada umum sudah terpasang bersamaan dengan KWH (*Kilo Watt Hour*) meter adalah MCB (*Miniatur Circuit Breaker*) dan sekering yang terpasang setelah KWH meter dimana jika terjadi gangguan pada instalasi baik disebabkan oleh hubung singkat atau beban lebih maka MCB akan memutuskan aliran listrik. Namun sensitifitas MCB berbeda – beda tergantung kualitas dari MCB tersebut. Apabila MCB tidak bekerja dengan baik maka jika terjadi gangguan maka sekering yang akan putus sehingga aliran listrik juga akan terputus dimana pada saat sekering putus maka harus diganti dengan yang baru untuk bisa menyambungkan kembali arus pada beban. Pada KWH meter yang pada umumnya terpasang tidak terdapat alat yang dapat menampilkan nilai dari tegangan, arus, *cosphi*, dan daya secara *real time* yang terpakai saat

itu dan juga tidak adanya alarm yang memberitahukan telah terjadi masalah seperti korsleting atau nilai arus berlebih pada beban yang terpasang.

Dari latar belakang diatas penulis berinisiatif membuat tugas akhir yang berjudul “Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega 16 dengan Sensor Arus” adalah sebuah alat proteksi yang digunakan untuk memproteksi dan mengukur tegangan, arus listrik dan juga daya listrik secara bersamaan. Alat ini juga dapat digunakan pada Genset yang belum memiliki alat monitor dimana energi listrik yang dibangkitkan dapat dimonitor nilai tegangan, arus, dan daya secara digital yang dialirkan oleh genset itu sendiri. Dengan dilengkapi sensor arus maka alat proteksi ini didesain selain untuk melindungi peralatan listrik apabila terjadi arus berlebih dan memutus hubungan listrik sekaligus berbunyinya alarm apabila ada perangkat yang berdaya besar.

Rancang bangun alat Sistem Proteksi Daya Listrik ini menggunakan mikrokontroler ATmega16 sebagai kendali dimana akan digunakan untuk mengolah sinyal *input* sehingga menjadi *ouput* yang berupa tampilan pada LCD dan akan mengendalikan *relay* sebagai pemutus hubungan listrik dan akan membunyikan *buzzer* apabila terjadi beban lebih dari yang ditentukan.

B. Identifikasi Masalah

Dari uraian latar belakang masalah diatas, maka dapat di-buat suatu identifikasi masalah sebagai berikut :

1. Belum banyak yang mengembangkan alat proteksi yang nilai proteksinya bisa ditentukan dan sekaligus sebagai alat ukur tegangan, arus, *cos phi*, dan daya yang menggunakan mikrokontroler ATmega 16.
2. Masih sedikit peralatan yang memanfaatkan mikrokontroler ATmega 16 sebagai kontrol proteksi yang memutus arus dengan cepat jika terjadi beban berlebih.
3. Belum adanya peralatan yang dikombinasikan antara alat proteksi, alat ukur, dan *alarm* peringatan beban berlebih yang berbasis mikrokontroler ATmega 16.
4. Alat ukur yang ada dipasaran memerlukan keahlian khusus dalam penggunaannya.

C. Batasan Masalah

Berdasarkan identifikasi masalah yang muncul perlunya pembatasan masalah sehingga ruang lingkup permasalahannya jelas. Dalam proyek akhir ini penulis membatasi masalah untuk membuat rancang bangun Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus sebagai berikut:

1. Sistem minimum menggunakan mikrokontroler ATmega16.
2. Bahasa pemrograman menggunakan Bahasa C.

3. Menggunakan sensor *Split Core Current Tranformator CT301*.
4. Pada *output* menggunakan LCD teks sebagai tampilan hasil pengukuran, *relay* sebagai pemutus arus dan *buzzer* sebagai alarm saat terjadi beban berlebih.
5. Terdapat *push button* yang digunakan untuk menentukan batas nilai arus.

D. Rumusan masalah

1. Bagaimana merancang *hardware* Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus?
2. Bagaimana merancang *software* Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus?
3. Bagaimana unjuk kerja Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus?

E. Tujuan

Pembuatan Proyek Akhir yang berjudul “Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus ” memiliki tujuan sebagai berikut:

1. Merealisasikan rancangan *hardware* Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus.
2. Merealisasikan rancangan *software* Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus.

3. Mengetahui unjuk kerja Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus.

F. Manfaat

Dalam pembuatan alat ini, di harapkan dapat memberikan manfaat sebagai berikut :

1. Bagi mahasiswa
 - a. Sebagai sarana implementasi pengetahuan yang di-
 - b. dapat di bangku pendidikan
 - c. Sebagai bentuk kontribusi terhadap Universitas baik dalam citra maupun daya tawar terhadap masyarakat luas.
2. Bagi Jurusan Pendidikan Teknik Elektronika
 - a. Terciptanya alat yang inovatif dan bermanfaat sebagai sarana ilmu pengetahuan.
 - b. Sebagai wujud partisipasi dalam pengembangan dibidang IPTEK
3. Bagi Dunia usaha dan Dunia Industri
 - a. Terciptanya Alat sebagai sarana peningkatan teknologi dalam dunia usaha dan dunia industri
 - b. Sebagai bentuk kontribusi terhadap industri dalam mewujudkan pengembangan teknologi.

G. Keaslian Gagasan

Sepengetahuan penulis, proyek akhir dengan judul “*Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega16 dengan Sensor Arus*” ini belum pernah diajukan untuk memperoleh gelar Ahli Madya atau gelar lainnya di suatu Perguruan Tinggi, dan sepanjang pengetahuan penulis juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Adapun yang menjadi ciri khas pada proyek akhir ini adalah :

1. Menggunakan sensor arus sebagai alat pendeteksinya.
2. Menggunakan ATmega16 sebagai sistem pengendalinya.
3. Menggunakan *relay* sebagai pemutus arus jika terjadi beban berlebih.
4. Menggunakan *buzzer* sebagai penanda apabila terjadi masalah.
5. Menggunakan *push button* sebagai penentu nilai arus.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Sistem Proteksi Daya Listrik

Sistem proteksi daya listrik adalah alat yang dibuat dari kombinasi alat proteksi, Alarm, dan alat ukur. Alat ini bekerja berdasarkan arus yang terdeteksi pada sistem sehingga didapatkan nilai arus yang digunakan sebagai acuan utama dimana nilai arus didapatkan dari sensor arus *Split Current Transformer CT301*.

Pada sistem proteksi nilai arus yang terdeteksi digunakan sebagai pembanding dimana jika nilai arus dari beban yang terpasang melebihi nilai arus yang ditentukan maka secara otomatis memutus arus sehingga beban yang terpasang tidak mendapat *supply* tegangan. Secara bersamaan *buzzer* akan berbunyi dimana fungsinya adalah sebagai alarm untuk memberitahukan beban yang terpasang telah melebihi batas arus yang ditentukan.

Keseluruhan sistem dibagi atas tiga bagian yaitu: 1. *Input* 2. Kendali, dan 3. *Output*.

Pada bagian *input* terdiri dari:

1. *Push button* yang berfungsi sebagai penentu nilai arus yang akan dibatasi.
2. Sensor arus *Split Current Transformer CT301* yang berfungsi sebagai pendeteksi nilai arus.

3. Pendeteksi beda fasa yang berfungsi sebagai pendeteksi nilai pergeseran beda fasa.

Bagian kendali terdiri dari mikrokontroler ATmega 16 yang berfungsi untuk mengendalikan keseluruhan sistem.

Sedangkan pada bagian *output* terdiri dari :

1. LCD 16x2 karakter sebagai media penampil hasil pengukuran nilai arus, tegangan, beda fasa, dan daya.
2. *Relay* berfungsi sebagai pemutus arus saat terjadi beban berlebih.
3. *Buzzer* berfungsi sebagai alarm saat terjadi beban berlebih.

B. Sensor

Sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut Transduser. (<http://id.wikipedia.org/wiki/Sensor>)

Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah besaran fisik (misalnya: temperatur, gaya, kecepatan putaran) menjadi besaran listrik yang proposional. Sensor dalam teknik pengukuran dan pengaturan ini harus memenuhi persyaratan-persyaratan kualitas yakni:

1. Linieritas

Konversi harus benar-benar proposional, jadi karakteristik konversi harus linier.

2. Tidak tergantung temperature

Keluaran konverter tidak boleh tergantung pada temperatur di sekelilingnya, kecuali sensor suhu.

3. Kepekaan

Kepekaan sensor harus dipilih sedemikian, sehingga pada nilai-nilai masukan yang ada dapat diperoleh tegangan listrik keluaran yang cukup besar.

4. Waktu tanggapan

Waktu tanggapan adalah waktu yang diperlukan keluaran sensor untuk mencapai nilai akhirnya pada nilai masukan yang berubah secara mendadak. Sensor harus dapat berubah cepat bila nilai masukan pada sistem tempat sensor tersebut berubah.

5. Batas frekuensi terendah dan tertinggi

Batas-batas tersebut adalah nilai frekuensi masukan periodik terendah dan tertinggi yang masih dapat dikonversi oleh sensor secara benar. Pada kebanyakan aplikasi disyaratkan bahwa frekuensi terendah adalah 0Hz.

6. Stabilitas waktu

Untuk nilai masukan (*input*) tertentu sensor harus dapat memberikan keluaran (*output*) yang tetap nilainya dalam waktu yang lama.

7. Histerisis

Gejala histerisis yang ada pada magnetisasi besi dapat pula dijumpai pada sensor. Misalnya, pada suatu temperatur tertentu sebuah sensor dapat memberikan keluaran yang berlainan. (<http://www.nubielab.com>)

C. Sensor Arus *Split Current Transformer CT301*

Transformator arus terdiri dari kumparan primer dan kumparan sekunder yang dililitkan pada suatu inti magnetik. Arus yang hendak dideteksi dialirkan ke kumparan primer. Arus ini menghasilkan suatu medan magnet yang mengimbas ke kumparan sekunder. Inti magnetik pada transformator berfungsi untuk membuat fluks magnetik yang dihasilkan kumparan primer sebanyak mungkin menembus kumparan sekunder. Perubahan fluks yang dihasilkan arus primer menyebabkan timbulnya tegangan listrik induksi pada kumparan sekunder. Arus yang dibangkitkan pada kumparan sekunder sebanding dengan arus primer, dan nilai kedua arus ini ditentukan oleh nilai jumlah lilitan masing-masing kumparan. Transformator arus memang menawarkan isolasi elektris, tetapi alat ini hanya bekerja untuk aplikasi arus bolak-balik (AC).



Gambar .1 Bentuk fisik Sensor Arus *Split Current Transformer CT301*

(Sumber : <http://www.zntar.com/37-269-large/ct301-split-core-current-transformer.jpg>)

Sensor *Split Current Transformer* menawarkan solusi berupa sensitivitas yang tinggi, ukuran yang kecil, dan reliabel. Selain itu piranti ini dapat digunakan untuk mendeteksi arus tanpa mengganggu rangkaian. Sensor *Split Current Transformer* bekerja berdasarkan prinsip diferensial dengan cara ini maka gangguan atau *noise* yang berasal dari lingkungan seperti temperatur atau pengaruh lingkungan lainnya bisa dihilangkan dan sensor dapat mengukur medan magnet yang sangat lemah. (<http://hgenius-electrical-eng.com/2011/03/sensor-arus.html>)

Specifications

split current transformer

1. ISO9001 & CE
2. High accuracy $\geq 0.5\%$
3. Excellent phase and ratio error
- 4 Various choice

split current transformer:

SPECIFICATION

Product Name	split current transformer
Model No.	CT301
Rated Current	75A
Rated Frequency	1KHz
Accuracy Class	1.0, 2.0
Current Ratio	3000:1
Load Resistance	100Ω
Central Hole Diameter	10mm/0.39"
Overall Size	26 x 23 x 36mm/1.02" x 0.91" x 1.42"(L*W*H)
Shell Material	Plastic
Color	Black
Frequency Band	50Hz~60Hz 1KHz MAX
Dielectric strength	3,000Vrms/ 2 seconds between secondary and primary
Saturation Current	$\geq 85A$ 50Hz
DCR	$510 \pm 10\% \Omega$
Inductance	$\geq 15H$ (@50HZ,1V)

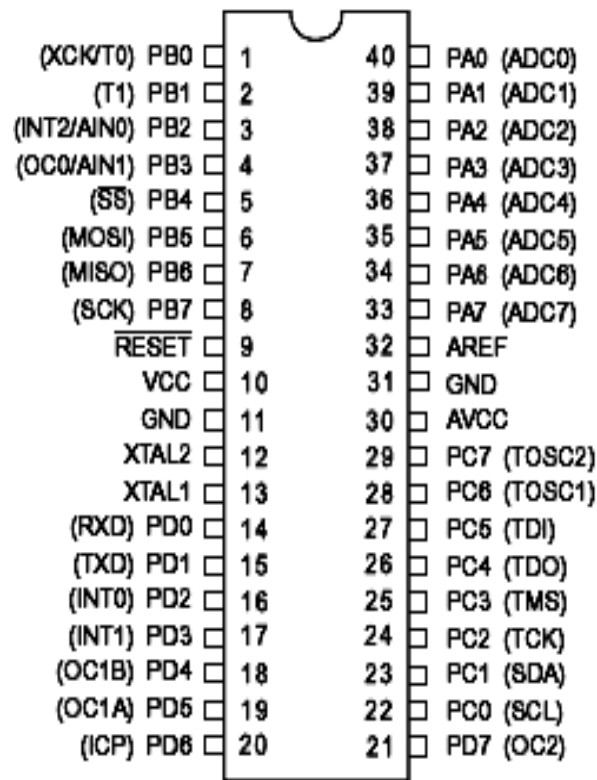
Gambar .2 Spesifikasi Sensor *Split Current Transformer CT301*

(Sumber : http://www.alibaba.com/product-detail/CT301-Mini-Black-split-current-transformer_262035426.html)

D. Mikrokontroler AVR (*Alf and Vegaard's Risc Processor*) ATmega 16

Mikrokontroler AVR adalah *single chip computer* yang memiliki kemampuan untuk diprogram dan digunakan untuk tugas-tugas yang berorientasi kontrol. Mikrokontroler datang dengan dua alasan utama, yang pertama adalah kebutuhan pasar (*market need*) dan yang kedua adalah perkembangan teknologi baru. Saat ini keluarga mikrokontroler yang ada di pasaran yaitu intel 8048 dan 8051(MCS51), Motorola 68HC11, *microchip* PIC, Hitachi H8, dan Atmel AVR. (Heri Andrianto 2008 : 1)

Secara teknis, hanya ada 2 macam mikrokontroler. Pembagian ini didasarkan pada kompleksitas instruksi-instruksi yang dapat diterapkan pada mikrokontroler tersebut. Pembagiannya yaitu RISC dan CISC. RISC (*Reduced Instruction Set Computer*). Instruksi yang dimiliki terbatas, tetapi memiliki fasilitas yang lebih banyak. CISC (*Complex Instruction Set Computer*). Instruksi bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya. Mikrokontroler *Alv and Vegard's RISC processor* atau sering disingkat AVR merupakan mikrokontroler RISC 8 bit. Karena RISC inilah sebagian besar kode instruksinya dikemas dalam satu siklus clock. AVR adalah jenis mikrokontroler yang paling sering dipakai dalam bidang elektronika dan instrumentasi. Secara umum, AVR dapat dikelompokkan dalam 4 kelas. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral dan fungsinya. Keempat kelas tersebut adalah keluarga ATTiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. IC mikrokontroler dikemas (*packaging*) dalam bentuk yang berbeda. Namun pada dasarnya fungsi kaki yang ada pada IC memiliki persamaan. Gambar salah satu bentuk IC seri mikrokontroler AVR ATmega16 dapat dilihat berikut.



Gambar 3. Konfigurasi PIN ATmega16

(Sumber: Data Sheet ATMEL No. Rev.2466T-AVR-07/10)

Berikut adalah penjelasan fungsi tiap kaki dari AVR ATmega16 :

1. *Port A* : Merupakan *8-bit bi-directional port I/O*. Setiap *pin*nya dapat menyediakan internal *pull-up* resistor (dapat diatur per bit). *Output buffer Port A* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port A* (DDRA) harus *disetting* terlebih dahulu sebelum *Port A* digunakan. Bit-bit DDRA diisi 0 jika ingin memfungsikan *pin-pin port A* yang bersesuaian sebagai *input*, atau diisi 1

jika sebagai *output*. Selain itu, kedelapan *pin port A* juga digunakan untuk masukan sinyal analog bagi *A/D converter*

2. *Port B* : Merupakan 8-bit *bi-directional port I/O*. Setiap *pinnya* dapat menyediakan internal *pull-up* resistor (dapat diatur per bit). *Output buffer Port B* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port B (DDRB)* harus diatur terlebih dahulu sebelum *Port B* digunakan. Bit-bit *DDRB* diisi 0 jika ingin memfungsikan *pin-pin port B* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. *Pin-pin port B* juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut:

Tabel 1. Fungsi-fungsi khusus *Port B* pada mikrokontroler ATmega16

Port Pin	Fungsi Khusus
PB0	T0 = timer/counter 0 external counter input
PB1	T1 = timer/counter 0 external counter input
PB2	AIN0 = analog comparator positive input
PB3	AIN1 = analog comparator negative input
PB4	SS = SPI slave select input
PB5	MOSI = SPI bus master output / slave input
PB6	MISO = SPI bus master input / slave output
PB7	SCK = SPI bus serial clock

3. *Port C* : Merupakan 8-bit *bi-directional port I/O*. Setiap *pin*nya dapat menyediakan internal *pull-up* resistor (dapat diatur per bit). *Output buffer Port C* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port C (DDRC)* harus *disetting* terlebih dahulu sebelum *Port C* digunakan. *Bit-bit DDRC* diisi 0 jika ingin memfungsikan *pin-pin port C* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, dua *pin port C* (PC6 dan PC7) juga memiliki fungsi alternatif sebagai *oscillator* untuk *timer/counter 2*.
4. *Port D* :Merupakan 8-bit *bi-directional port I/O*. Setiap *pin*nya dapat menyediakan internal *pull-up* resistor (dapat diatur per bit). *Output buffer Port D* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port D (DDRD)* harus *disetting* terlebih dahulu sebelum *Port D* digunakan. *Bit-bit DDRD* diisi 0 jika ingin memfungsikan *pin-pin port D* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin* pada *port D* juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

Tabel 2. Fungsi-fungsi khusus *Port D* pada mikrokontroler ATmega16

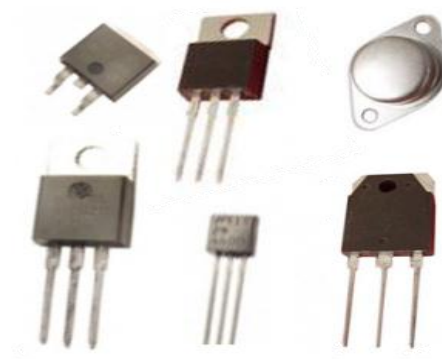
Port Pin	Fungsi Khusus
PD0	RDX (<i>UART input line</i>)
PD1	TDX (<i>UART output line</i>)
PD2	INT0 (<i>external interrupt 0 input</i>)
PD3	INT1 (<i>external interrupt 1 input</i>)
PD4	OC1B (<i>Timer/Counter1 output compareB match output</i>)
PD5	OC1A (<i>Timer/Counter1 output compareA match output</i>)
PD6	ICP (<i>Timer/Counter1 input capture pin</i>)
PD7	OC2 (<i>Timer/Counter2 output compare match output</i>)

5. RESET merupakan *pin* yang digunakan untuk me-reset mikrokontroler
6. XTAL1 dan XTAL 2 merupakan *pin* masukan *clock* eksternal.
7. AVcc merupakan kaki masukan tegangan untuk ADC.
8. AREF merupakan *pin* masukan tegangan referensi ADC.
9. Vcc merupakan *pin* yang berfungsi sebagai masukan catu daya.
10. GND merupakan *pin Ground*. (Heri Adrianto 2008 : 5-7)

E. Transistor

Transistor adalah alat semikonduktor yang dipakai sebagai penguat, sebagai sirkuit pemutus dan penyambung (*switching*), stabilisasi tegangan, modulasi sinyal atau sebagai fungsi lainnya. Transistor dapat berfungsi

semacam kran listrik, dimana berdasarkan arus *inputnya* (BJT) atau tegangan *inputnya* (FET), memungkinkan pengaliran listrik yang sangat akurat dari sirkuit sumber listriknya. (<http://id.wikipedia.org/wiki/Transistor>)



Gambar 4. Bentuk fisik Transistor

(sumber:<http://wartadigital.com/wpcontent/uploads/2012/transistor>)

Pada umumnya, transistor memiliki 3 terminal, yaitu Basis (B), *Emitor* (E) dan *Kolektor* (C). Tegangan yang disatu terminal misalnya *Emitor* dapat dipakai untuk mengatur arus dan tegangan yang lebih besar daripada arus *input* Basis, yaitu pada keluaran tegangan dan arus *output* *Kolektor*.

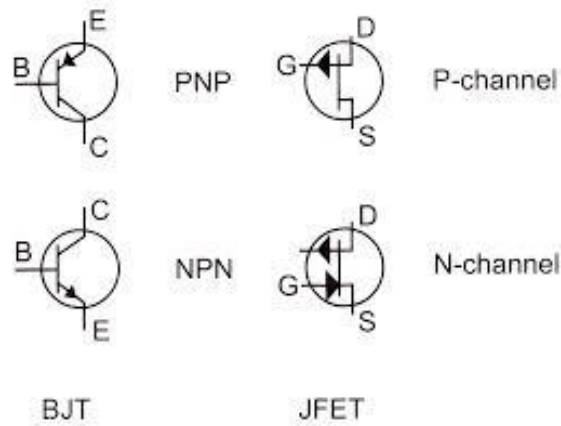
Transistor merupakan komponen yang sangat penting dalam dunia elektronik *Modern*. Dalam rangkaian analog, transistor digunakan dalam amplifier (penguat). Rangkaian analog melingkupi pengeras suara, sumber listrik stabil (stabilisator) dan penguat sinyal radio. Dalam rangkaian-rangkaian digital, transistor digunakan sebagai saklar berkecepatan tinggi.

Beberapa transistor juga dapat dirangkai sedemikian rupa sehingga berfungsi sebagai *logic gate*, memori dan fungsi rangkaian-rangkaian lainnya.

Dari banyak tipe-tipe transistor *Modern*, pada awalnya ada dua tipe dasar transistor, *Bipolar Junction Transistor* (BJT atau transistor bipolar) dan *Field-Effect Transistor* (FET), yang masing-masing bekerja secara berbeda. Transistor bipolar dinamakan demikian karena kanal konduksi utamanya menggunakan dua polaritas pembawa muatan: elektron dan lubang, untuk membawa arus listrik. Dalam BJT, arus listrik utama harus melewati satu daerah/lapisan pembatas dinamakan *depletion zone*, dan ketebalan lapisan ini dapat diatur dengan kecepatan tinggi dengan tujuan untuk mengatur aliran arus utama tersebut.

FET (juga dinamakan transistor unipolar) hanya menggunakan satu jenis pembawa muatan (elektron atau *Hole*, tergantung dari tipe FET). Dalam FET, arus listrik utama mengalir dalam satu kanal konduksi sempit dengan *depletion zone* di kedua sisinya (dibandingkan dengan transistor bipolar dimana daerah Basis memotong arah arus listrik utama). Dan ketebalan dari daerah perbatasan ini dapat diubah dengan perubahan tegangan yang diberikan, untuk mengubah ketebalan kanal konduksi tersebut.

1. Jenis Transistor



Gambar 5. Simbol Transistor

(Sumber : <http://gemeltelkomsp2.webs.com/komponen-elektronika>)

Secara umum, transistor dapat dibeda-bedakan berdasarkan banyak kategori:

- Materi semikonduktor: Germanium, Silikon, Gallium Arsenide
- Kemasan fisik: *Through Hole Metal*, *Through Hole Plastic*, *Surface Mount*, IC, dan lain-lain
- Tipe: UJT, BJT, JFET, IGFET (MOSFET), IGBT, HBT, MISFET, VMOSFET, MESFET, HEMT, SCR serta pengembangan dari transistor yaitu IC (*Integrated Circuit*) dan lain-lain.
- Polaritas: NPN atau *N-channel*, PNP atau *P-channel*
- Maximum kapasitas daya: *Low Power*, *Medium Power*, *High Power*

- f) Maximum frekwensi kerja: *Low*, *Medium*, atau *High* Frequency, RF transistor, *Microwave*, dan lain-lain.

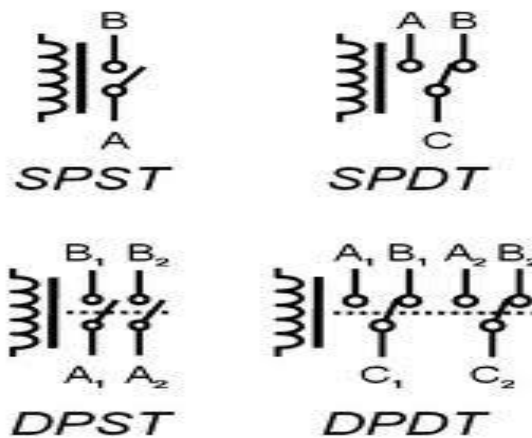
2. BJT

BJT (*Bipolar Junction Transistor*) adalah salah satu dari dua jenis transistor. Cara kerja BJT dapat dibayangkan sebagai dua dioda yang terminal positif atau negatifnya berdempet, sehingga ada tiga terminal. Ketiga terminal tersebut adalah emiter (E), *kolektor* (C), dan basis (B).

Perubahan arus listrik dalam jumlah kecil pada terminal basis dapat menghasilkan perubahan arus listrik dalam jumlah besar pada terminal *kolektor*. Prinsip inilah yang mendasari penggunaan transistor sebagai penguat elektronik. Rasio antara arus pada kolektor dengan arus pada basis biasanya dilambangkan dengan β atau h_{FE} . β biasanya berkisar sekitar 100 untuk transistor-transistor BJT.

F. Relay

Relay adalah saklar elektronik yang didasarkan atas elektrik dan mekanik. Kontrol elektrik diterapkan untuk mendapatkan gerakan mekanik. Ada banyak tipe *relay* yang konstruksinya juga berbeda tergantung jenis kontakannya. Gambar di bawah ini menunjukkan berbagai jenis simbol *relay*. (<http://elektronika-dasar.web.id/teori-relay>)



Gambar 6. Simbol relay

(Sumber : <http://commons.wikimedia.org>)

1. *Relay SPST (Single Pole SingleThrough)*

Relay dengan satu induk saklar dengan satu saluran kontak (*normally closed*).

2. *Relay DPST (Double Pole SingleThrough)*

Sama seperti SPST tetapi mempunyai dua buah saklar terpisah yang bekerjanya serentak/bersamaan dan satu saluran kontak (*normally closed*) untuk tiap saklar.

3. *Relay SPDT (Single Pole Double Through)*

Merupakan *relay* yang mempunyai satu induk saklar untuk menghubungkan dua saluran kontak (*normally closed dan normally open*) yang dihubungkan bergantian.

4. *Relay DPDT (Double Pole Double Through)*

Sama seperti SPDT tetapi mempunyai dua buah saklar terpisah yang bekerja serentak dan dua saluran kontak (*normally closed* dan *normally open*) untuk tiap saklar.

G. *Transformer (Trafo)*

Transformer (trafo) adalah alat yang digunakan untuk menaikkan atau menurunkan tegangan bolak-balik (AC). *Transformer* terdiri dari 3 komponen pokok yaitu: kumparan pertama (primer) yang bertindak sebagai input, kumparan kedua (skunder) yang bertindak sebagai output, dan inti besi yang berfungsi untuk memperkuat medan magnet yang dihasilkan.

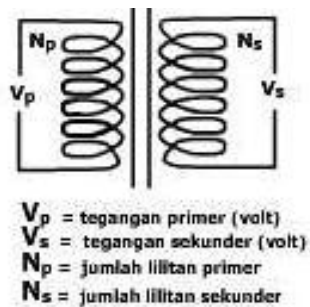
(<http://www.elektronika123.com/trafo-atau-transformator/>)



Gambar 7. Bentuk fisik Trafo

(Sumber : <http://all-elektro.blogspot.com/2012/01/transformator-trafo.html>)

Prinsip kerja dari sebuah *transformator* adalah sebagai berikut, ketika kumparan primer dihubungkan dengan sumber tegangan bolak-balik, perubahan arus listrik pada kumparan primer menimbulkan medan magnet yang berubah. Medan magnet yang berubah diperkuat oleh adanya inti besi dan diantarkan inti besi ke kumparan sekunder, sehingga pada ujung-ujung kumparan sekunder akan timbul ggl induksi. Efek ini dinamakan induktansi timbal-balik (*mutual inductance*).



Gambar 8. Hubungan kumparan primer dan sekunder trafo
 (Sumber : <http://all-elektro.blogspot.com/2012/01/transformator-trafo.html>)

Hubungan antara tegangan primer, jumlah lilitan primer, tegangan sekunder, dan jumlah lilitan sekunder, dapat dinyatakan dalam persamaan:

$$\frac{V_p}{V_s} = \frac{N_p}{N_s}$$

V_p = tegangan primer (volt)

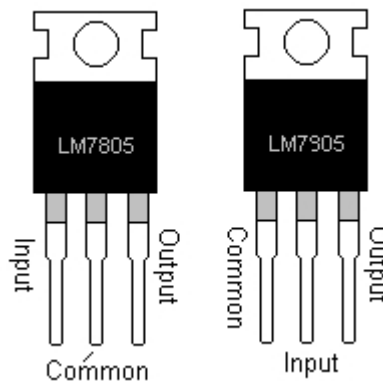
V_s = tegangan sekunder (volt)

N_p = jumlah lilitan primer

N_s = jumlah lilitan sekunder

H. IC Regulator

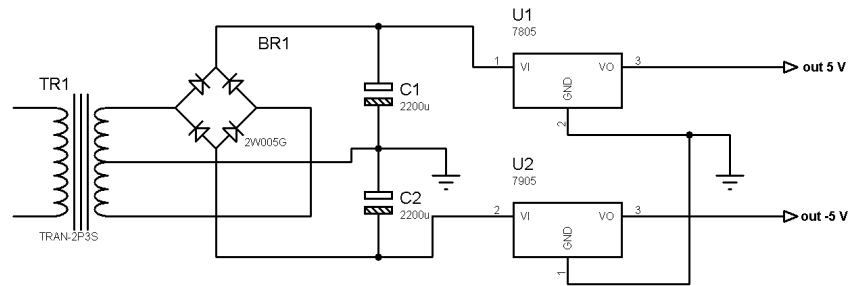
IC ini berfungsi sebagai pembatas dan penstabil tegangan yang keluar dari rangkaian penyearah. Terdapat dua jenis *IC Regulator* yaitu sebagai regulator tegangan positif dan sebagai regulator tegangan negatif. Perbedaannya terletak dari seri IC tersebut, untuk tegangan positif mempunyai seri 78XX sedangkan pada regulator tegangan negatif 79XX. Akhiran XX menunjukkan batas tegangan yang keluar dari IC tersebut. Sebagai contoh IC dengan seri 7805 berarti merupakan regulator tegangan positif dengan keluaran 5 V. (<http://teknikelektronika.com/jenis-ic-voltage-regulator-pengatur-tegangan/>)



Gambar 9. Susunan kaki *IC Regulator*

(Sumber : <http://www.ucontrol.com.ar>)

Skema rangkaian penggunaan *IC regulator* pada *power supply* sebagai pembatas dan penstabil tegangan dapat dilihat pada gambar dibawah ini :



Gambar 10. Rangkaian *power supply* dengan *IC regulator*

I. *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*).

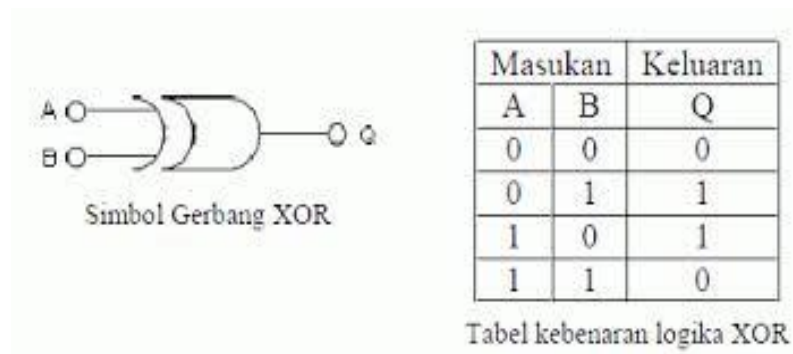


Gambar. 11 Bentuk fisik *buzzer*

(Sumber : www.futurlec.com)

J. Gerbang XOR

Gerbang XOR (*exclusive-or*) akan memberikan keluaran 1 jika masukan-masukannya mempunyai keadaan yang berbeda. Dari gambar dibawah dapat dilihat keluaran logika pada gerbang XOR..



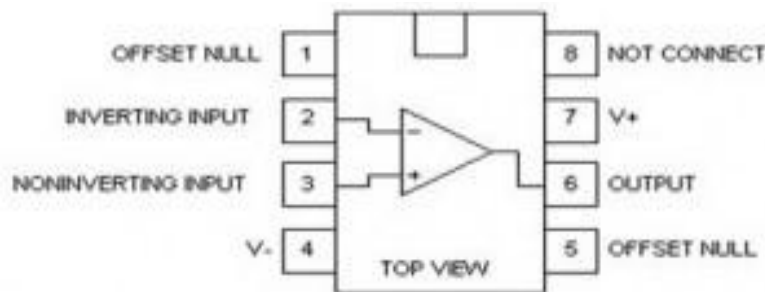
Gambar 12. Gerbang XOR 2 input

(Sumber: <https://encryptedtbn2.gstatic.com/images?q=tbn:ANd9GcThGvEiXlar dVIXrSixVliYTvmDwDhSsag28oWOCrrVnxIDuZDuVQ>)

K. Penguat Operasional (*Op-Amp*)

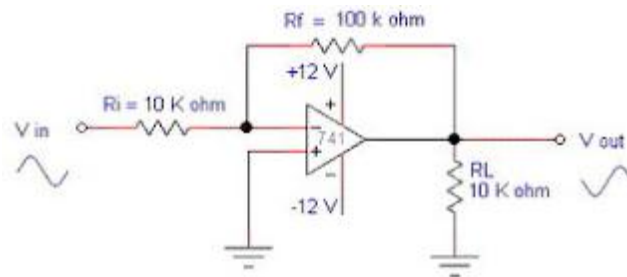
Operational Amplifier atau di singkat *op-amp* merupakan salah satu komponen analog yang sering digunakan dalam berbagai aplikasi rangkaian elektronika. Aplikasi *op-amp* yang paling sering dipakai antara lain adalah rangkaian *inverter*, *non-inverter*, *integrator* dan *differensiator*.

Pada pembahasan ini penulis menggunakan *op-amp* sebagai penguat *inverting* sebagai penguatan dari output sensor arus. Salah satu tipe *Op-Amp* yang populer adalah LM741. Pada IC ini terdapat dua pin input, dua pin power supply, satu pin output, satu pin NC (No Connection), dan dua *pin offset null*. Pin *offset null* memungkinkan kita untuk melakukan sedikit pengaturan terhadap arus internal di dalam IC untuk memaksa tegangan *output* menjadi nol ketika kedua *input* bernilai nol.



Gambar 13. Konfigurasi pin IC LM 741

Penguat *Op-Amp* membalik (*inverting*) akan menguatkan tegangan pada masukan serta membalik hasil penguatan tersebut, sehingga keluaran dari rangkaian ini akan selalu memiliki polaritas yang berlawanan dengan sinyal masukannya.



Gambar 14. Rangkaian penguat *inverting*

Penguatan tegangan pada rangkaian ini di tentukan menurut

$$A_v = -\frac{R_f}{R_i}$$

Tegangan keluaran diperoleh dengan jalan mengalikan tegangan masukan yang diketahui dengan faktor penguatan, atau

$$V_{out} = -\frac{R_f}{r_i} V_{in}$$

Tanda minus diabaikan dalam perhitungan karna hanya menunjukkan bahwa keluaran berlawanan fasa terhadap masukannya.

(<http://www.ilmu.8k.com/pengetahuan/opamp.html>.)

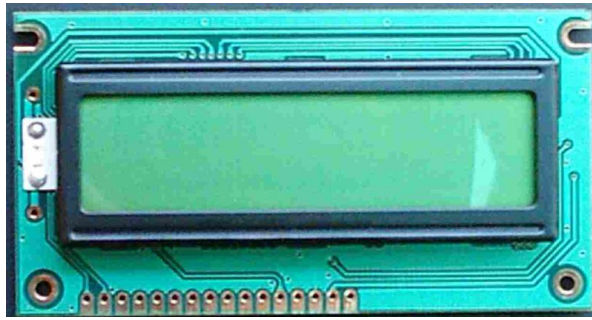
L. LCD Display 16x2 M1632

LCD merupakan alat penampil karakter yang dapat berupa angka, huruf atau bisa juga gambar. Jenis-jenis LCD yang ada dipasaran ada dua jenis yaitu LCD teks dan LCD grafik. LCD yang digunakan adalah jenis LCD M1632. Modul M1632 merupakan modul LCD dengan tampilan 16x2 baris dengan konsumsi daya yang rendah. LCD Teks adalah jenis LCD yang

digunakan untuk menampilkan teks atau angka dalam kode ASCII. Sedangkan *Graphic* LCD dapat menampilkan gambar karena setiap sel dari grafik LCD memuat "*dot*" kristal cair. LCD banyak digunakan sebagai *display* dari alat – alat elektronika seperti kalkulator, *multitester digital*, jam *digital* dan sebagainya. (Syamsul Rizal 2011:1)

Tabel 3. Konfigurasi PIN LCD 16x2

Fungsi	Pin	deskripsi	Data Logika	Keterangan
Ground	1	VSS (Ground)	-	0V
Tegangan sumber	2	VDD (VCC)	-	+5V
Kontras	3	VEE	-	0-Vdd
Kendali LCD	4	RS	0 1	
	5	R/W	0 1	
	6	E	0 1 floating	
Data	7	D0	0/1	Bit 0
	8	D1	0/1	Bit
	9	D2	0/1	Bit
	10	D3	0/1	Bit
	11	D4	0/1	Bit
	12	D5	0/1	Bit
	13	D6	0/1	Bit
	14	D7	0/1	Bit
Anoda	15	backlight, lampu background	-	
Katoda	16	backlight, lampu background	-	



Gambar 15. Bentuk fisik LCD M1632

(Sumber : http://www.commonswikimedia.org/wiki/File:Lcd_16x2.png)

M. Perangkat lunak

a) Bahasa C

Bahasa C adalah bahasa mesin tingkat tinggi dimana dapat dengan mudah untuk melakukan pemrograman terhadap mikrokontroler. Secara umum pemrograman mikrokontroler terdiri atas empat blok, setiap blok tersebut mempunyai definisi tersendiri yaitu :

1. *Header*
2. Deklarasi konstanta global
3. Fungsi dan atau prosedur (biasa dibawah program utama)
4. Program utama

Secara umum, pemrograman C paling sederhana dapat dilakukan dengan hanya menuliskan program utamanya saja. Beberapa peraturan yang ada dalam bahasa C adalah:

(a) Header

Header berisi include file (.h), yaitu *library* (pustaka) yang akan digunakan dalam pemrograman. Perhatikan contoh dibawah ini:

```
#include <mega16.h>

#include <delay.h>

#include <stdio.h>
```

(b) Tipe Data

Berikut ini adalah tabel tipe-tipe variabel data yang dapat digunakan di *compiler Code Vision AVR*:

Tabel 4. Berbagai macam tipe data dalam *compiler Code Vision AVR*

Type	Size (Bits)	Range
Bit	1	0,1
Char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
Int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
Float	32	$\pm 1.175e-38$ to $\pm 3.402e38$
Double	32	$\pm 1.175e-38$ to $\pm 3.402e38$

(c) Konstanta

Penulisan konstanta adalah sebagai berikut:

- (1) *Integer* atau *long integer* dapat ditulis dengan format decimal (contoh 1234), biner dengan awalan 0b (contoh 0b101001), heksadesimal dengan awalan 0x (contoh 0xff) atau octal dengan awalan 0 (contoh 0777).
- (2) *Unsigned integer* ditulis dengan diakhiri U (contoh 10000U).
- (3) *Long integer* ditulis dengan diakhiri L (contoh 99L).
- (4) *Unsigned long integer* ditulis dengan diakhiri UL (contoh 99UL).
- (5) *Floating poin* ditulis dengan diakhiri F (contoh 1.234F).

Karakter konstanta harus ditulis dalam tanda kutip (contoh 'a'), sedangkan konstanta *string* harus dalam tanda kutip dua (contoh "Saya Belajar C").

(d) Label, Variabel, Fungsi

Identifikasi label, variabel dan fungsi dapat berupa huruf (A...Z, a...z) Dan angka (0...9), juga karakter *underscore* (_). Meskipun begitu identifikasi hanya bias dimulai dengan huruf atau karakter *underscore*.

Yang lebih penting lagi, identifikasi ini *Case is significant*, yaitu huruf besar dan kecil berbeda. Misal Variable1 tidak sama dengan variabel1. Identifikasi bisa memuat sebanyak 32 karakter.

(e) Komentar

Komentar diawali dengan tanda `'/*'` dan diakhiri dengan `'*/'`.

Perhatikan contoh dibawah:

```
/* ini komentar */
```

(f) *Reserved Keywords*

Berikut ini adalah daftar kata baku yang tidak bisa dipakai (*reserved keywords*) untuk label, identifikasi atau variable:

Break	flash	signed	do	int	typedef
Bib	float	sizeof	double	interrupt	union
Case	for	sfrb	eeprom	long	unsigned
Char	funcused	sfrw	else	register	void
const	goto	static	enum	return	volatile
continue	if	struct	extern	short	while
default	inline	witch			

(g) Operator

Suatu intruksi pasti mengandung operator dan operand. Operand adalah variabel atau konstanta yang merupakan bagian pernyataan sedangkan operator adalah suatu simbol yang menyatakan operasi mana yang akan dilakukan oleh operand tersebut.

Sebagai Contoh: $c = a + b$;

Ada tiga operand (a, b dan c) dan dua operator (= dan +). Operator dalam

C dibagi menjadi 3 kelompok, yaitu :

(1) *unary* operator yang beroperasi pada satu operand, misal: $-n$

(2) *binary* operator yang beroperasi pada dua operand, misal: $a - n$

(3) *ternary* operator memerlukan tiga / lebih operand, misal: $a = (b * c) + d$

(h) Aritmatika

Tabel 5. Simbol dan Aritmatika

simbol	Contoh	aritmatika
+	$c = a + b$ $n = n + 2$	penjumlahan
-	$c = a - b$ $n = n - 2$	pengurangan
++	++i	Kenaikan (increment), sama dengan $i = i + 1$
--	--i	Penurunan (decrement), sama dengan $i = i - 1$
*	$c = a * b$	perkalian
/	$c = a / b$	pembagian
	$n = n / 2$	
%	Sisa = $a \% b$	Menghasilkan sisa dari pembagi. a dan b bilangan bulat
=	$a = b$	Pemberian nilai
+=	$A += 2$	Penambahan suatu nilai pada suatu variabel yang sudah ada sebelumnya. Sama dengan $a = a + 2$
-=	$A -= 2$	Pengurangan suatu nilai pada suatu variabel yang sudah ada sebelumnya. Sama dengan $a = a - 2$
*=	$A *= 2$	Pengalian suatu nilai pada suatu variabel yang sudah ada sebelumnya. Sama dengan $a = a * 2$
/=	$a /= 2$	Pembagian suatu nilai pada suatu variabel yang sudah ada sebelumnya. Sama dengan $a = a / 2$
%=	$A \% = 2$	Sisa dari suatu nilai pada suatu variabel yang sudah ada sebelumnya yang dibagi oleh nilai atau variable lainnya. Sama dengan $a = a \% 2$
*	*pointer	Menunjukkan isi dari pointer

(i) **Logika**

Tabel 6. Simbol dan Pembandingan

Simbol	Contoh	Logika pembandingan
<code>==</code>	If (a == b)	Logika sama dengan, digunakan untuk pembandingan. Menghasilkan nilai <i>true</i> jika a=b
<code>!=</code>	If (a != b)	Tidak sama dengan. Menghasilkan nilai <i>true</i> jika a ≠ b.
<code><</code>	If (a < b)	Logika lebih kecil dari. Menghasilkan nilai <i>true</i> jika a < b.
<code><=</code>	If (a <= b)	Logika lebih kecil sama dengan dari. Menghasilkan nilai <i>true</i> jika a ≤ b.
<code>></code>	If (a > b)	Logika lebih besar dari. Menghasilkan nilai <i>true</i> jika a > b.
<code>>=</code>	If (a >= b)	Logika lebih besar sama dengan dari. Menghasilkan nilai <i>true</i> jika a ≥ b.
<code>!</code>	If (!a)	NOT
<code>&&</code>	If (a == b && a == c)	AND
<code> </code>	If (a == b a == c)	OR

(j) **Manipulasi Bit**

Tabel 7. Manipulasi Bit

<code>~</code>	A = ~ b	<i>Complement.</i> b = 1100; a = 0011.
<code>&</code>	c = a & b	AND untuk manipulasi <i>bit</i> . a = 1100; b = 1001; c = 1000.
<code> </code>	c = a b	OR untuk manipulasi bit. a = 1100; b = 1001; c = 0101.
<code><<</code>	c = a << n	<i>Shift left</i> , manipulasi bit menggeser kekiri sejauh n bit a = 1101; n = 2; maka c = 110100.
<code>>></code>	c = a >> n	<i>Shift Right</i> , manipulasi bit menggeser kekanan sejauh n bit a = 11010; n = 2; c = 0110.

(k) Percabangan

(1) *if – then*

bentuk umum dari percabangan ini adalah:

```
if (kondisi) {  
  // pernyataan  
};
```

(2) *if – then – else*

bentuk umum dari percabangan ini adalah:

```
if (kondisi)  
{  
  // pernyataan a  
}  
else  
{  
  // pernyataan b  
};
```

(3) *switch – case*

Pernyataan *switch – case* digunakan jika terjadi banyak percabangan. Pada uraian di bawah ini merupakan contoh dari yang termasuk dalam penjelasan. Struktur penulisan pernyataan ini adalah sebagai berikut:

```
switch (ekspresi)  
{  
  pernyataan1  
  break;  
  case konstanta2:  
    pernyataan2  
    break;  
  case konstantaN:  
    pernyataanN  
    break;  
}
```

(4) *switch – case – default*

Pernyataan *switch – case – default* hampir sama dengan *switch – case*. Hal yang membedakannya adalah bahwa dengan adanya *default* maka jika tidak terdapat kondisi *case* yang sesuai dengan ekspresi *switch* maka akan menuju pernyataan yang terdapat pada bagian *default*.

```
switch (ekspresi)
{
  case konstanta1
  break;
  case konstanta2:
  pernyataan2
  break;
  .....
  case konstantaN:
  pernyataanN
  break;
  default:
  Pernyataan-pernyataan;
}
```

(5) *Perulangan for*

Pernyataan *for* akan melakukan perulangan berapa kali sesuai yang diinginkan. Struktur penulisan perulangan *for* adalah sebagai berikut:

```
For (mulai ; kondisi ; penambahan atau pengurangan)
{
  Pernyataan-pernyataan;
};
```

(6) *while*

Bentuk dari perulangan *while* adalah sebagai berikut:

```
while (kondisi)
```

```
{  
pernyataan-pernyataan;  
}
```

(7) *do – while*

Bentuk perulangan ini kebalikan dari *while – do*, yaitu pernyataan dilakukan terlebih dahulu kemudian diuji kondisinya

```
do  
{  
pernyataan-pernyataan;  
} while (kondisi);
```

(l) **Konversi pola (%)**

Karakter % dipakai sebagai operator konversi pola. Konversi pola akan sangat berguna pada saat kita menampilkan hasil ke LCD.

a. %d menampilkan bilangan bulat positif.

Contoh: `Sprintf(buf,"angka%d",14);`

b. %o menampilkan bilangan oktal bulat.

c. %x menampilkan bilangan heksadesimal bulat.

d. %u menampilkan bilangan desimal tanpa tanda.

e. %f menampilkan bilangan pecahan.

f. %i menampilkan bilangan *integer*.

g. %c menampilkan karakter yang ditunjukkan bilangan ASCII.

(m) Prosedur

Prosedur adalah suatu kumpulan instruksi untuk mengerjakan suatu keperluan tertentu tanpa mengembalikan suatu nilai.

```
....  
void nama_prosedur (parameter1, parameter2,.....parameterN)  
{  
}  
....
```

(n) Fungsi

Fungsi adalah suatu kumpulan instruksi untuk mengerjakan suatu keperluan tertentu dengan hasil akhir pengembalian nilai dari keperluan tersebut.

```
....  
Type data nama_fungsi (parameter1, parameter2,  
...parameterN)  
{  
Pernyataan-pernyataan;  
return variable_hasil;  
}  
....
```

(o) Memasukkan Bahasa *Assembly*

Sering disebut juga dengan *in-line assembly*. Pemrograman dengan bahasa C ini masih dapat memasukkan bahasa *assembly* ke dalam program C. yaitu:

```
...  
#asm //dimulai dengan #asm  
nop // blok bahasa assembly  
nop //  
#endasm // diakhiri dengan #endasm
```

... •

Atau jika hanya beberapa instruksi maka kita bisa melakukannya dengan

cara:

... •

```
#asm("nop\nop\nop")
```

b) Code Vision AVR

CodeVisionAVR merupakan sebuah *cross-compiler C*, *Integrated Development Environment (IDE)*, dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. *CodeVisionAVR* dapat dijalankan pada sistem operasi *Windows 95, 98, Me, NT4, 2000, dan XP*.

Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded*. File *object COFF* hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan C, dengan pengamatan variabel, menggunakan *debugger* Atmel AVR Studio. IDE mempunyai fasilitas internal berupa *software AVR Chip In-System Programmer* yang memungkinkan untuk melakukan transfer program kedalam *chip* mikrokontroler setelah sukses melakukan kompilasi/assembly secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan *Atmel STK500/AVRISP/AVRProg*, *Kanda Systems STK200+/300*, *Dontronics DT006*, *Vogel Elektronik VTEC-ISP*, *Futurlec JRAVR* dan *MicroTronics*

ATCPU/Mega2000programmers/development boards. Untuk keperluan *debugging* sistem *embedded*, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah terminal. (Syaiful Alam: 2010)

CodeVisionAVR juga mempunyai *Automatic Program Generator* bernama *CodeWizardAVR*, yang memungkinkan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

1. *Set-up* akses memori eksternal
2. Identifikasi sumber reset untuk chip
3. Inisialisasi port *input/output*
4. Inisialisasi interupsi eksternal
5. Inisialisasi *Timer/Counter*
6. Inisialisasi *Watchdog-Timer*
7. Inisialisasi *UART (USART)* dan komunikasi serial
8. Inisialisasi Pembanding Analog
9. Inisialisasi ADC
10. Inisialisasi Antarmuka SPI
11. Inisialisasi Antarmuka *Two-Wire*
12. Inisialisasi Antarmuka CAN
13. Inisialisasi Bus I2C, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307
14. Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20
15. Inisialisasi modul LCD

BAB III

KONSEP RANCANGAN

Dalam melakukan perancangan alat Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus. Metode yang digunakan adalah sebagai berikut: identifikasi kebutuhan, analisis kebutuhan, perancangan perangkat keras dan perangkat lunak, pembuatan serta pengujian alat. Berikut metode rancang bangun secara detail.

A. Identifikasi Kebutuhan

Dalam merealisasikan perangkat Sistem Proteksi Daya Listrik berbasis Mikrokontroler ATmega 16 dengan Sensor Arus. Hal yang harus dilakukan terlebih dahulu melakukan identifikasi kebutuhan sistem diantaranya:

1. Unit catu daya
2. Unit sensor tegangan
3. Sensor arus
4. Unit penguat instrumentasi
5. Unit detektor beda fasa
6. Unit sistem minimum
7. Unit penampil LCD 16x2
8. *Push button*
9. *Relay*
10. *Buzzer*

B. Analisis Kebutuhan

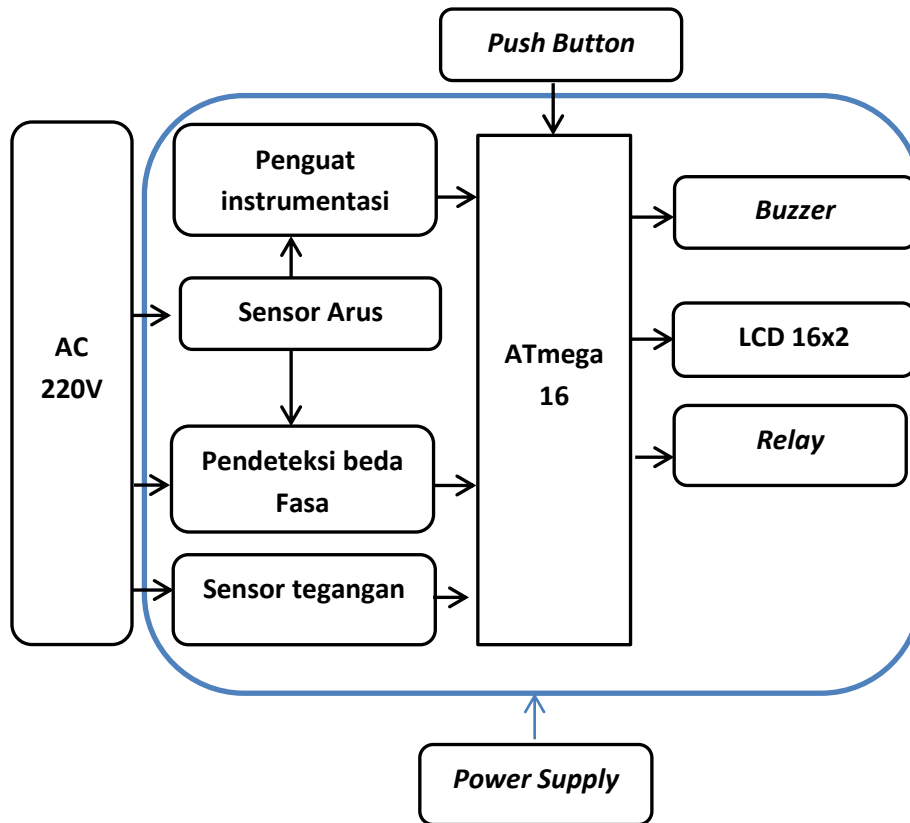
Berdasarkan identifikasi kebutuhan tersebut maka diperoleh beberapa analisis kebutuhan yaitu :

1. Rangkaian catu daya yaitu catu daya simetris dengan keluaran tegangan $\pm 8V$.
Catu daya tunggal $+5V$ dan $+12V$.
2. Rangkaian sensor tegangan menggunakan trafo *step down*.
3. Sensor arus *Split Current Transformer CT301*.
4. Rangkaian penguat instrumentasi menggunakan IC *Op-amp* LM 358.
5. Rangkaian detektor beda fasa menggunakan IC *Op-amp* LM 358 dan LM3209.
6. Rangkaian sistem minimum menggunakan mikrokontroler ATmega 16.
7. LCD karakter 16 x2 sebagai media penampil dalam memberi informasi hasil pengukuran dan menu pengaturan dalam sistem.
8. Menggunakan *Push button* sebagai masukan menu pengaturan untuk menambah atau mengurangi nilai arus yang akan diolah pada sistem.
9. *Relay* sebagai pemutus arus saat terjadi beban berlebih.
10. *Buzzer* sebagai alarm saat terjadi beban berlebih.

C. Perancangan Sistem

1. Perancangan perangkat keras (*Hardware*)

Berdasarkan analisis kebutuhan diperlukan suatu susunan blok diagram sebagai berikut:



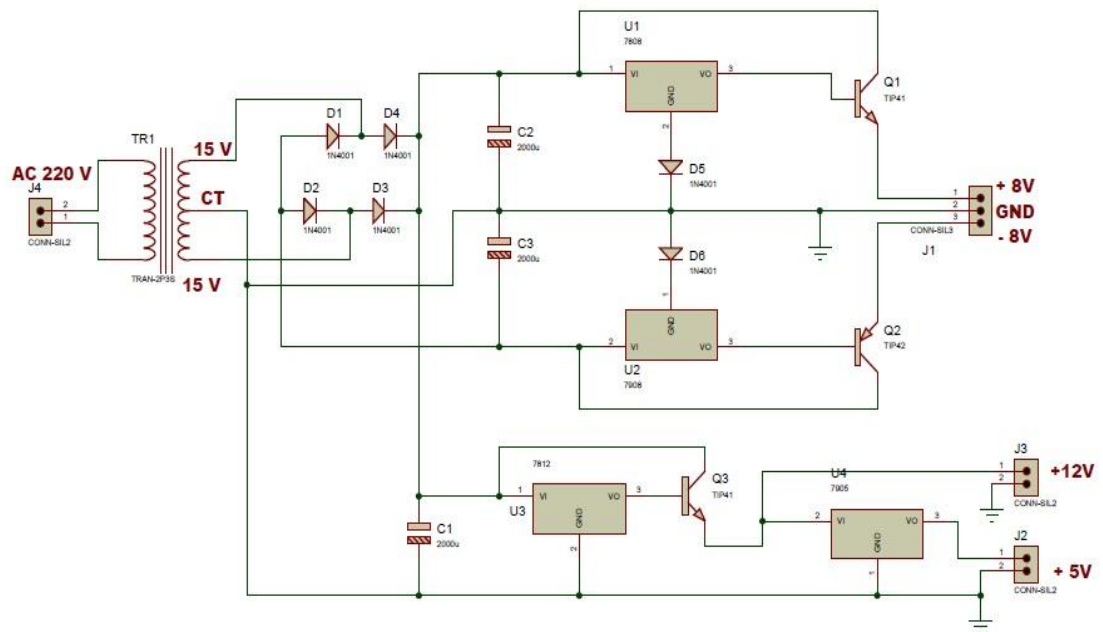
Gambar 16. Blok diagram perancangan alat

Alat ini menggunakan sensor arus *Split Current Transformer CT301* sebagai sensor utama selain sensor arus terdapat juga rangkaian sensor tegangan dan rangkaian pembaca nilai faktor daya. Pada rangkaian mikrokontroler yang menggunakan ATmega 16 nantinya akan membaca nilai arus, tegangan, dan nilai Faktor daya. Dimana nilai keluaran dari sensor arus *Split Current Transformer CT301*, rangkaian sensor tegangan, dan rangkaian pendeteksi nilai faktor daya akan diolah pada mikrokontroler dan kemudian dari pembacaan nilai arus, tegangan, dan faktor daya maka akan didapatkan nilai daya. Daya yang dibaca dalam sistem ini adalah daya nyata.

Keluaran dari sistem ini adalah ditampilkannya nilai arus, tegangan, faktor daya, dan daya nyata pada LCD 16x2. Selain itu keluaran dari sistem ini terdapat juga *relay* yang akan aktif apabila pada terjadi beban lebih pada sistem yang dihubungkan ke beban sekaligus mengaktifkan *buzzer* sebagai alarm.

a. Rangkaian Catu Daya Simetris dan Tunggal

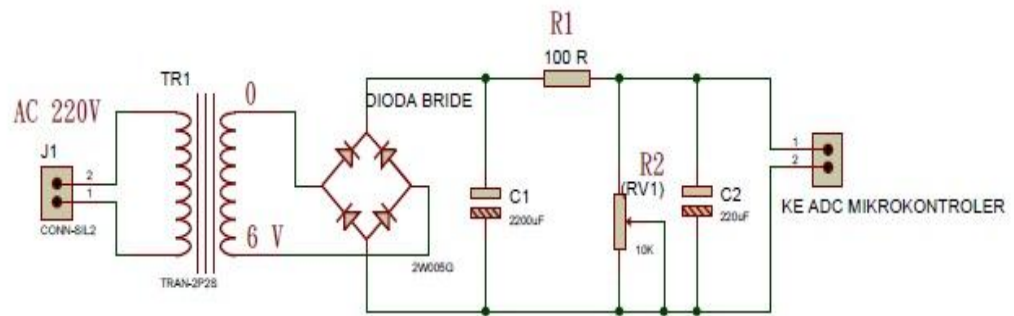
Catu daya mutlak dibutuhkan dalam setiap perangkat elektronika. Pada perancangan alat ini, penulis menggunakan jenis catu daya simetris. Catu daya simetris merupakan catu daya yang dapat mengeluarkan tegangan positif dan negatif. Tegangan yang di butuhkan untuk men-*supply* rangkaian adalah sebesar 5V DC untuk sistem minimum ATmega 16. Tegangan 12 V DC digunakan pada rangkaian *output relay dan buzzer* sedangkan tegangan $\pm 8V$ digunakan untuk men-*supply* rangkaian penguat instrumentasi dan rangkaian penentu beda fasa. Pada catu daya ini menggunakan *IC regulator 7805, 7808 dan 7812* sebagai *output* tegangan positif dan *IC 7908* sebagai output tegangan negatif. Berikut ini gambar dari rangkaian *power supply* dalam proyek ini:



Gambar 17. Rangkaian *Power Supply Unit*

b. Rangkaian Sensor Tegangan menggunakan Trafo *Step-down*

Sensor tegangan menggunakan trafo *step-down* ini difungsikan sebagai pembaca tegangan. Trafo *step-down* digunakan sebagai penurun tegangan AC, kemudian disearahkan menggunakan penyearah gelombang penuh dengan empat buah dioda dan *filter* kapasitor agar *noise* tidak terbaca pada ADC mikrokontroler. Pada proyek ini, trafo yang digunakan sebagai penurun tegangan sebesar 500 mA dengan mengambil terminal tegangan 0 dan 6 V. Gambar rangkaian blok penurun tegangan dapat dilihat pada gambar 18:



Gambar 18. Rangkaian Sensor Tegangan

Perhitungan perbandingan nilai resistor menggunakan persamaan :

$$V_o = \frac{R_2}{R_1 + R_2} \times V_{in}$$

Keterangan :

V_{in} = Tegangan input

V_{out} = Tegangan output

R_1 = Resistor posisi 1 pada gambar 12

R_2 = Resistor posisi 2 (RV 1) pada gambar 12

Diketahui :

$V_{in} = 6 \text{ V}$

$V_{out} = 5 \text{ V}$

$R_1 = 100 \text{ ohm}$

Ditanya R2 =...?

Penyelesaian

$$V_o = \frac{R_2}{R_1 + R_2} \times V_{in}$$

$$5 = \frac{R_2}{100 + R_2} \times 6$$

$$5 = \frac{6R_2}{100 + R_2}$$

$$6R_2 = 5 \times (100 + R_2)$$

$$6R_2 = 500 + 5R_2$$

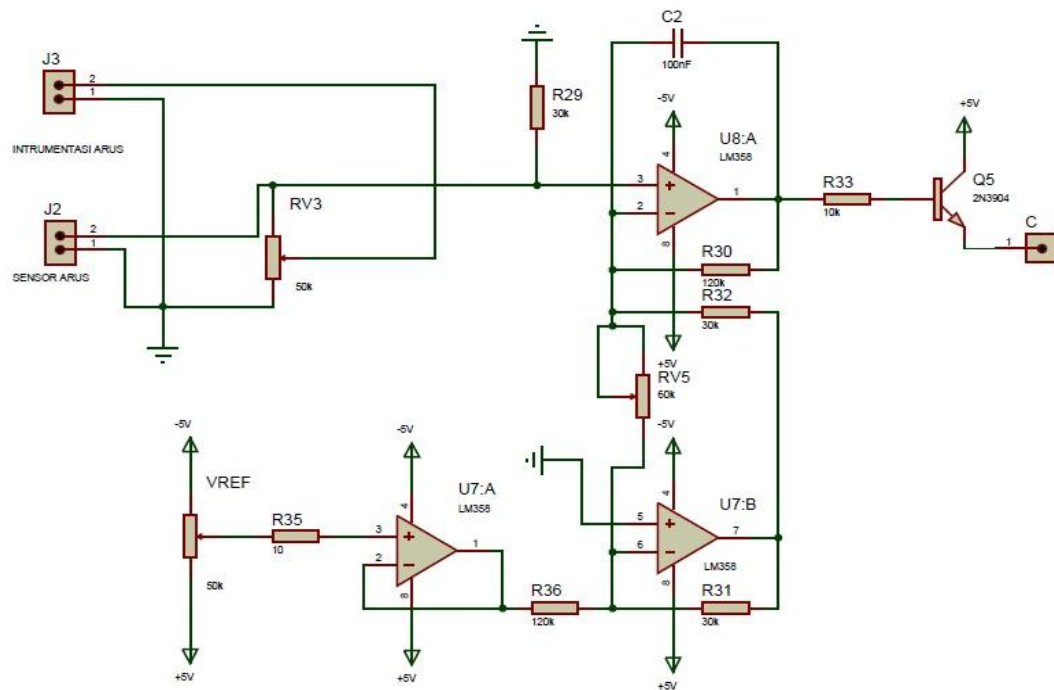
$$R_2 = 500 \text{ ohm}$$

Agar tegangan keluar dapat diatur untuk tujuan kalibrasi, maka R2 diganti menggunakan potensio sebesar 10 kΩ.

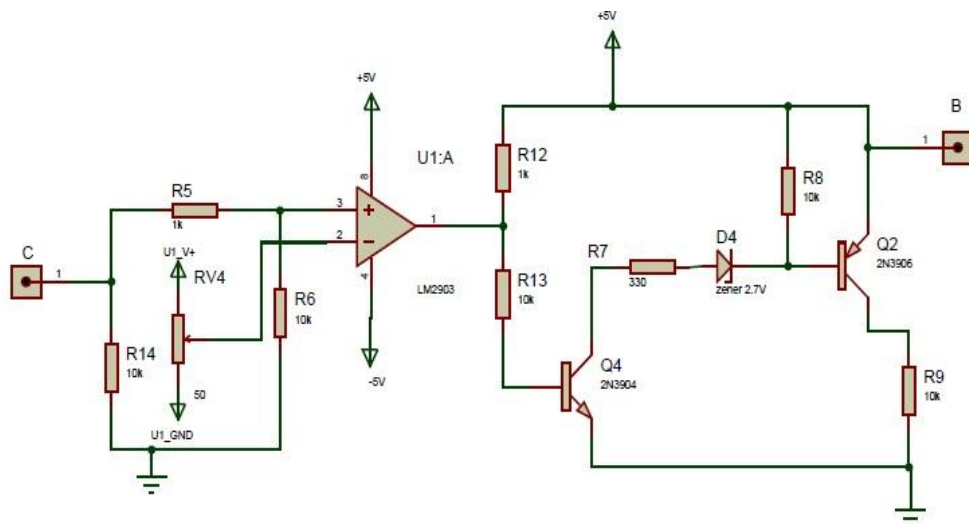
c. Rangkaian Penguat Instrumentasi

Rangkaian penguat instrumentasi ini berfungsi sebagai penguat tegangan dari keluaran sensor arus yang kecil sehingga keluaran dapat dibaca oleh ADC. IC yang digunakan yaitu tipe *Op-amp* LM 358. Gambar dari rangkain penguat instrumentasi dapat dilihat pada gambar dibawah ini:

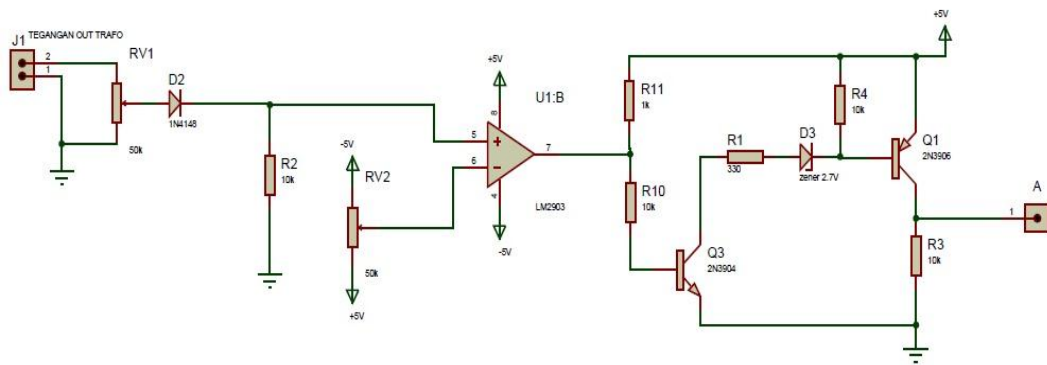
gerbang logika AND 7408, OR 74312, dan XOR 7486. Rangkaian dari detektor beda fasa dapat dilihat pada gambar dibawah ini:



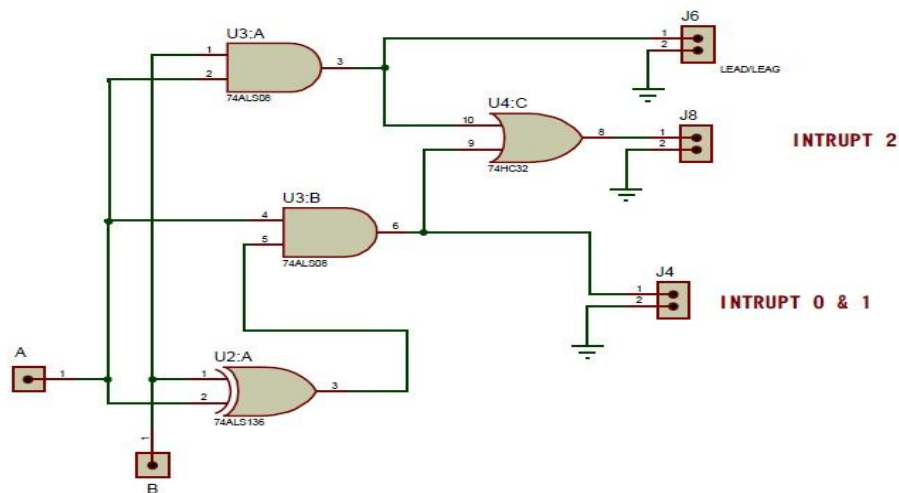
Gambar 20. Rangkaian penguat Instrumentasi Beda fasa



Gambar 21. Rangkaian Komparator Beda Fasa Arus



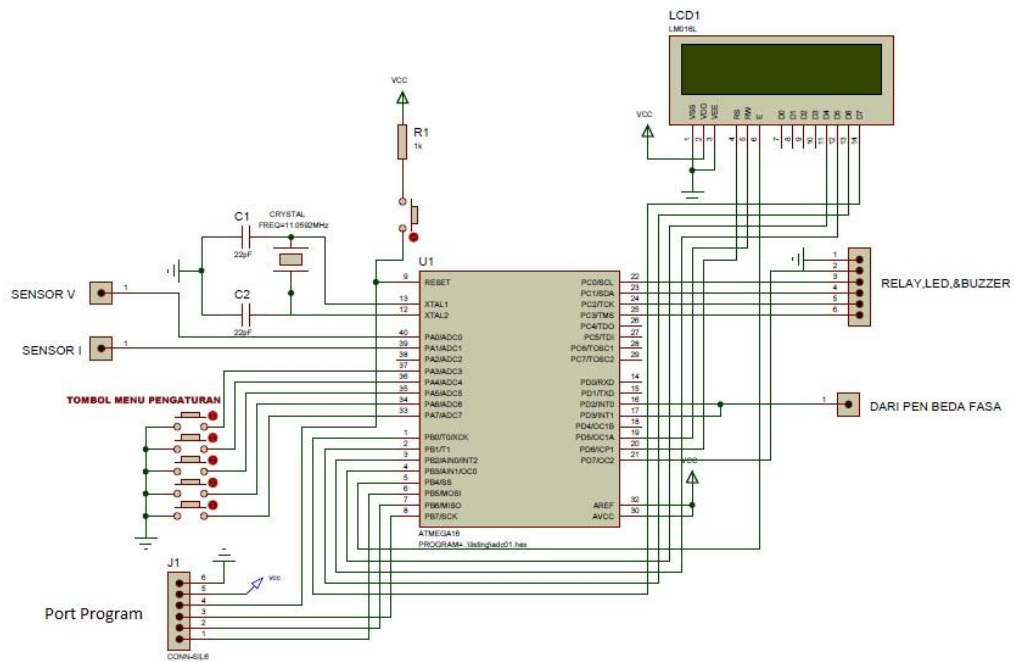
Gambar 22. Rangkaian Penguat Komparator Beda Fasa Tegangan



Gambar 23. Rangkaian logika pemroses Sinyal Beda Fasa Tegangan terhadap Arus

e. Rangkaian sistem minimum Mikrokontroler ATmega 16

Rangkaian sistem minimum mikrokontroler ATmega 16 merupakan rangkaian utama dalam proses pengolahan sinyal *input*. Rangkaian ini memproses semua sinyal *input* baik dari sensor arus, sensor tegangan, dan detektor beda fasa. Sinyal - sinyal input kemudian diproses untuk ditampilkan pada LCD. PIN dari ATmega 16 yang digunakan untuk mengolah masing-masing masukan adalah dua *pin* ADC dan satu *pin* Interrupt, *pin-pin* tersebut masing-masing akan berfungsi



Gambar 25. Konfigurasi kaki LCD pada *PIN* ATmega 16

g. *Push button*

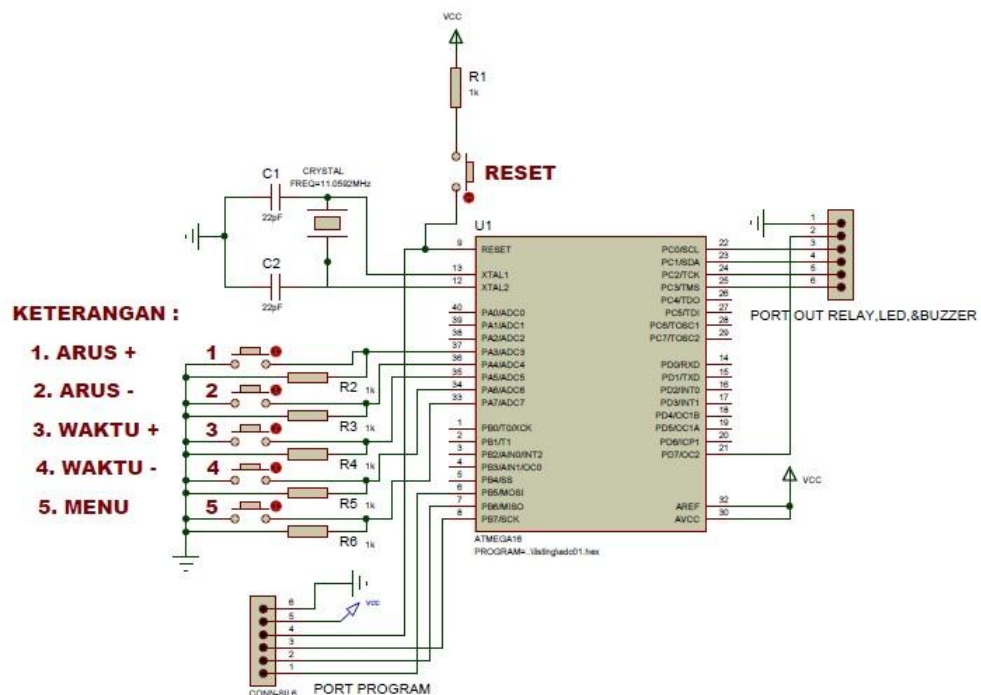
Push button digunakan untuk menentukan batas nilai arus yang akan dibatasi pada beban sehingga alat dapat mendeteksi berapa nilai arus pada beban yang dikehendaki. Terdapat lima buah *push button* yang satu buah *push button* digunakan menjadi *push button* menu. Dua *push button* digunakan menentukan naik atau turun batas nilai arus sedangkan dua *push button* lainnya digunakan untuk menentukan naik atau turun lamanya waktu proteksi. *Push button* menu dibagi menjadi tiga yaitu 1. Menu penampil hasil pengukuran, 2. Menu batas nilai arus, dan 3. Menu waktu proteksi.

Fungsi dari masing – masing menu adalah sebagai berikut :

- 1) Menu penampil hasil pengukuran berfungsi untuk menampilkan hasil pengukuran nilai arus (I), tegangan (V), faktor daya (PF), dan daya (P).

- 2) Menu batas nilai arus berfungsi untuk menentukan nilai arus yang akan dibatasi pada beban.
- 3) Menu waktu proteksi berfungsi untuk menentukan lama waktu proteksi terhadap beban yang terpasang yang dibuat dalam satuan menit.

Rangkaian dari *push button* pada sistem minimum mikrokontroler ATmega 16 dapat dilihat pada gambar dibawah ini :



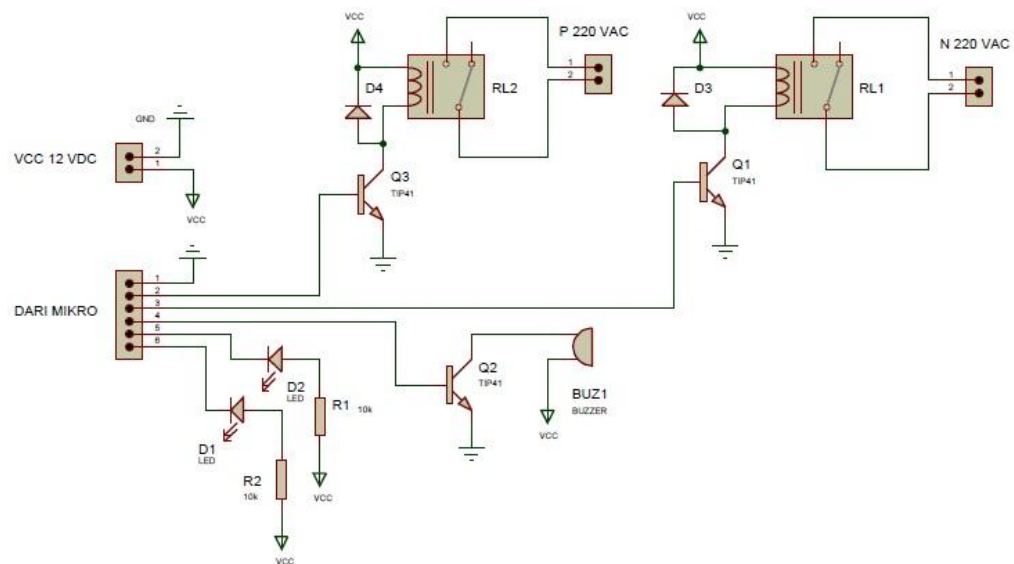
Gambar 26. Rangkaian *Push Button*

h. Rangkaian *Relay* dan *Buzzer*

Rangkaian *relay* dan *buzzer* adalah rangkaian output yang berfungsi sebagai pemutus arus dan *buzzer* juga akan berbunyi saat terjadi beban berlebih pada beban. Untuk mengaktifkan koil *relay* dan *buzzer* dibutuhkan transistor TIP41 untuk memperbesar arus sehingga dapat mengaktifkan koil *relay* dan *buzzer* karena *output* mikrokontroler tidak mampu mengeluarkan arus yang dibutuhkan

relay dan *buzzer*. Rangkaian ini juga terdapat dioda yang berfungsi agar tidak ada efek induksi dari *relay* yang dapat mengganggu pembacaan pada ADC maupun *Interrupt*.

Rangkaian *relay* dan *buzzer* dapat dilihat seperti pada gambar dibawah ini :



Gambar 27. Rangkaian *Relay* dan *Buzzer*

2. Perancangan Perangkat Lunak (*Software*)

Dalam perancangan alat perangkat lunak sangatlah dibutuhkan. Alat berbasis mikrokontroler tidak akan dapat bekerja tanpa adanya perangkat lunak di dalamnya yang akan menjalankan sistem. Perancangan perangkat lunak tersebut memerlukan beberapa langkah kerja yang sistematis, sesuai bahasa pemrograman yang digunakan. Perancangan perangkat lunak dimulai dari penentuan langkah kerja yang digambarkan dengan algoritma dan diagram alir (*flowchart*). Tujuan membangaun program dengan membuat algoritma dan *flowchart* terebih dahulu dalah agar dapat diperoleh langkah-langkah yang paling efektif dan efisien.

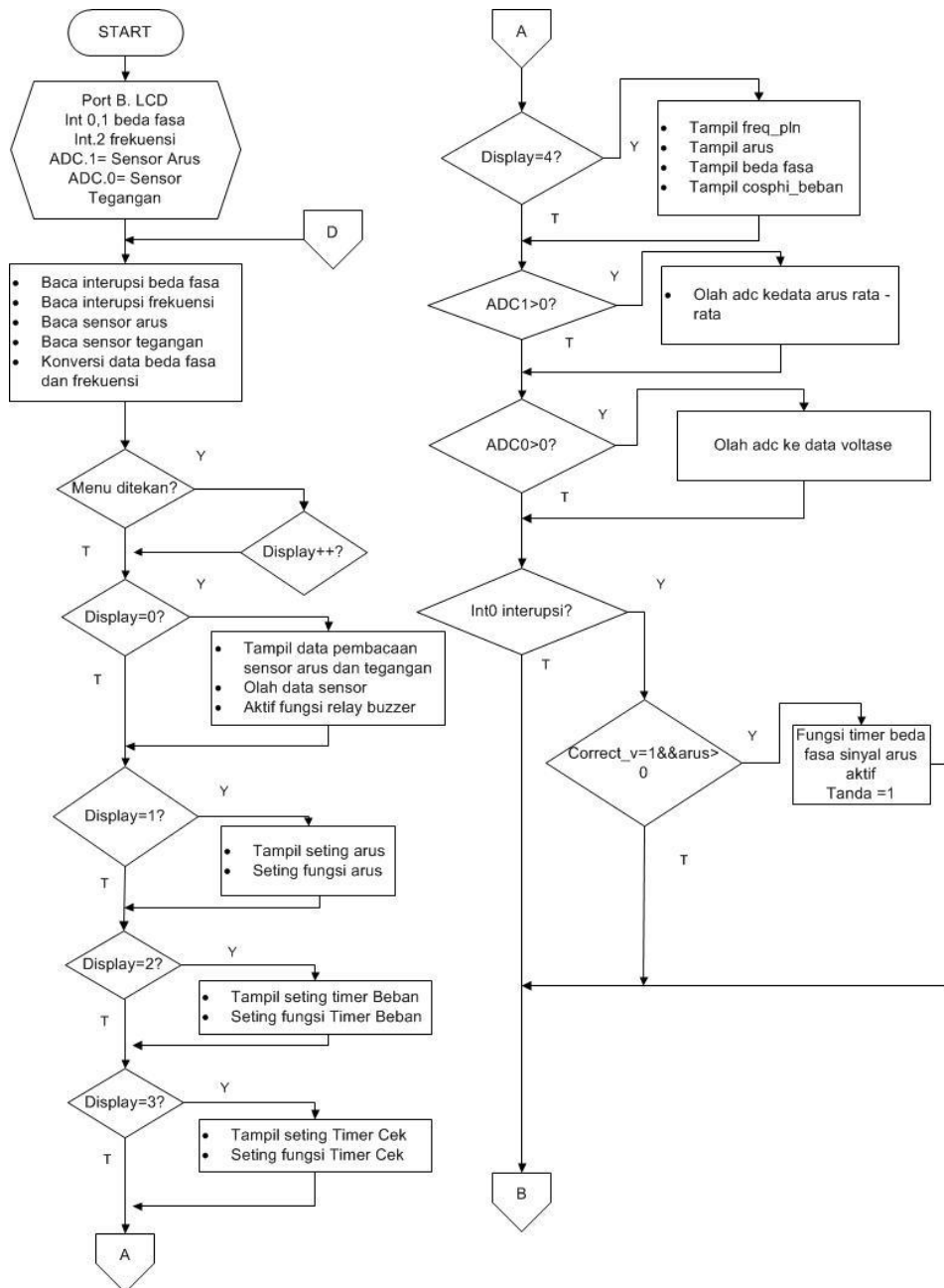
(a) Algoritma

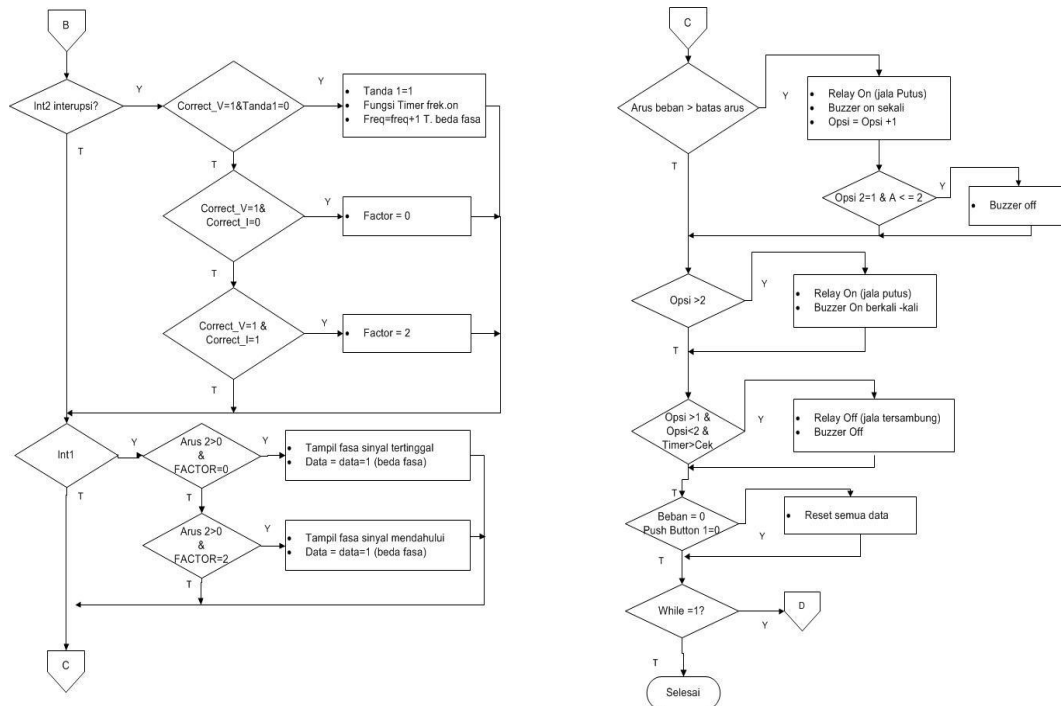
- Step 1* : Mulai.
- Step 2* : Inisialisasi Port B.LCD, intrupt 0&1 beda fasa, intrupt 2 frekuensi,timer 16bit, ADC.1 sensor Arus ADC.0 Sensor Tegannngan.
- Step 3* : Baca intrupsi beda fasa, intrupsi frekuensi, sensor arus, tegangan, Konversi data beda fasa dan frekuensi.
- Step 4* : Tombol menu ditekan jika ya display = display + 1jika tidak lanjut ke step berikutnya
- Step 5* : Apakah display = 0 jika ya tampilkan data pembacaan sensor dan tegangan, olah data sensor, aktifkan fungsi *relay* dan *buzzer* jika tidak lanjut ke step selanjutnya.
- Step 6* : Apakah display = 1 jika ya tampilkan setting arus dan fungsi setting arus jika tidak lanjut ke step selanjutnya
- Step 7* : Apakah display = 2 jika ya tampilkan setting timer beban dan fungsi setting timer beban jika tidak lanjut ke step selanjutnya
- Step 8* : Apakah display = 3 jika ya tampilkan set timer cek dan fungsi setting timer cek tidak lanjut ke step selanjutnya
- Step 9* : Apakah display = 4 jika ya tampilkan frekuensi pln, arus, beda fasa, dan cosphi beban jika tidak lanjut ke step selanjutnya
- Step 10* : Apakah ADC 1 > 0 jika ya olah data ADC kedata arus rata – rata jika tidak lanjut ke step selanjutnya

- Step 11* : Apakah $ADC\ 0 > 0$ jika ya olah data ADC ke data voltase jika tidak lanjut ke step selanjutnya
- Step 12* : Apakah $int\ 0$ ada intrupsi jika ya lanjut ke step berikutnya jika tidak lanjut ke step 14
- Step 13* : Apakah $correct_V=1$ & $arus\ 2 > 0$ jika ya fungsi timer beda fasa sinyal arus aktif Tanda=1 jika tidak lanjut ke step 14
- Step 14* : Apakah $int\ 2$ ada intrupsi jika ya lanjut ke step berikutnya jika tidak lanjut ke step 18
- Step 15* : Apakah $correct_V=1$ tanda1=0 jika ya tanda 1=1 fungsi timer $freq\ On\ freq = freq + 1$ T beda fasa jika tidak lanjut ke step selanjutnya
- Step 16* : Apakah $correct_V=1$ & $correct_I=0$ jika ya factor = 0 jika tidak lanjut ke step selanjutnya
- Step 17* : Apakah $correct_V=1$ & $correct_I=1$ jika ya factor =2 jika tidak lanjut ke step 18
- Step 18* : Apakah $Int\ 1$ berintrupsi jika ya lanjut ke step berikutnya jika tidak lanjut ke step 21
- Step 19* : Apakah $arus\ 2 > 0$ & factor = 0 jika ya tampilkan fasa sinyal tertinggal $data = data + 1$ (T beda fasa) jika tidak lanjut ke step berikutnya
- Step 20* : Apakah $arus\ 2 > 0$ jika ya tampilkan fasa sinyal mendahului data $=data+1$ (T beda fasa) jika tidak lanjut ke step 21

- Step 21* : Apakah arus > batas arus jika ya maka relay On (tegangan Jala PLN beban putus) buzzer On sekali, opsi=opsi+1 jika tidak lanjut ke step23
- Step 22* : Apakah opsi $2 < 1$ & $A < 2$ jika ya maka buzzer off jika tidak lanjut ke step 23
- Step 23* : Apakah opsi >2 jika ya maka relay On (tegangan Jala PLN beban putus) Buzzer on berkali-kali jika tidak lanjut ke step berikutnya
- Step 24* : Apakah opsi > 1 & opsi < 2 & timer > cek jika ya maka buzzer Off, relay off (tegangan jala PLN ke beban tersambung) jika tidak lanjut ke step berikutnya
- Step 25* : Apakah beban = 0 push button 1 = 0 jika ya reset semua data jika tidak lanjut ke step berikutnya
- Step 26* : Apakah *while* =1 jika ya kembali ke step 3 jika tidak maka selesai
- Step 27* : Selesai

(b) *Flowchart program*





Gambar 28. Gambar *flowchart*

D. Pembuatan Alat

Dalam pembuatan Sistem Proteksi Daya Listrik berbasis mikrokontroler ATmega 16 dengan sensor arus terdapat beberapa alat yang akan digunakan yaitu :

1. Solder
2. Atraktor
3. Obeng set
4. Bor dan mata bor
5. Tang potong dan tang cucut

6. Multimeter
7. Gergaji besi
8. *Cutter*
9. Lem tembak
10. Gunting
11. Setrika
12. Unit komputer atau laptop
13. *Downloader*
14. Printer laser

Sedangkan bahan-bahan yang dibutuhkan antara lain

1. Komponen meliputi IC mikrokontroler ATmega16, Sensor arus *Split Current Transformer CT301*, Trafo 1A CT dan 500mA non CT, *Relay*, *Buzzer*, IC LM741, IC LF351, IC *regulator LM 7805, 7812, 7905* dan *7912*, LCD 16 x 2, *Push button*, *Crystal*, Transistor TIP41 dan komponen pendukung lainnya seperti resistor, kapasitor, dioda ,dan lain-lainnya.
2. PCB fiber polos
3. Timah
4. Pelarut FeCl
5. *Thinner*
6. *Kertas Art Paper*
7. Kabel- kabel
8. Lem bakar dan lem *acrylic*

9. Mur dan baut

10. Soket soket

11. Box *acrylic* dan kaki-kaki *box*

Berikut ini adalah tahap pembuatan alat

1. Pembuatan *Printed Circuit Board (PCB)*

A. Pembuatan *layout* PCB

Langkah pertama dalam pembuatan PCB adalah menggambar rangkaian menggunakan *software ISIS Proteus* kemudian di“*export*” ke *ARES* dibuat jalur PCBnya .

B. Penyablonan PCB

Setelah gambar rangkaian selesai dibuat langkah selanjutnya yaitu menyablon jalur PCBnya. Jalur PCB disablon pada PCB fiber polos.

Untuk proses penyablonan dilakukan dengan cara :

- 1) Mencetak jalur PCB pada kertas *Art Paper* dengan menggunakan Printer laser.
- 2) Membersihkan PCB yang akan digunakan terlebih dahulu.
- 3) Jalur PCB yang sudah dicetak pada kertas *Art Paper* selanjutnya disablonkan ke PCB dengan cara disetrika selama kurang lebih 5 menit.
- 4) Setelah gambar jalur PCB menempel pada PCB selanjutnya dibersihkan menggunakan air agar kertas dapat terpisah.
- 5) Setelah terpisah antara PCB dan kertas *Art Paper* maka akan terbentuk jalur PCB pada PCB.

2. Pelarutan dan pengeboran PCB

Proses selanjutnya yaitu melarutkan PCB dengan menggunakan cairan *feri chloride* sampai jalur rangkaian terbentuk, kemudian PCB diangkat dari cairan *feri chloride* tadi dan dibersihkan menggunakan cairan tiner agar bekas cat sablon dapat hilang selanjutnya dibersihkan dengan air. Langkah selanjutnya PCB dibor sesuai dengan titik – titik yang telah ditentukan.

3. Pengecekan Jalur Rangkaian

Setelah selesai dirakit maka dilakukan pengujian yaitu mengecek jalur apakah rangkaian tidak mengalami konsleting dan jalur dalam keadaan baik ketika dialiri arus listrik.

4. Pemasangan Komponen

Setelah PCB selesai dibor maka langkah selanjutnya yaitu melakukan pemasangan seluruh komponen sesuai rangkaian dengan urutan :

- a. Menyiapkan komponen - komponen yang dibutuhkan.
- b. Memasang komponen dari komponen pasif terlebih dahulu contohnya resistor kemudian baru komponen aktif seperti transistor. Dengan tujuan komponen aktif tidak rusak akibat panas berlebih saat penyolderan.
- c. Menyolder kaki komponen sampai semua komponen terpasang.
- d. Memotong kaki komponen agar rapi.

5. Pembuatan *Box* Rangkaian

1. Perencanaan ukuran

Panjang : 22cm

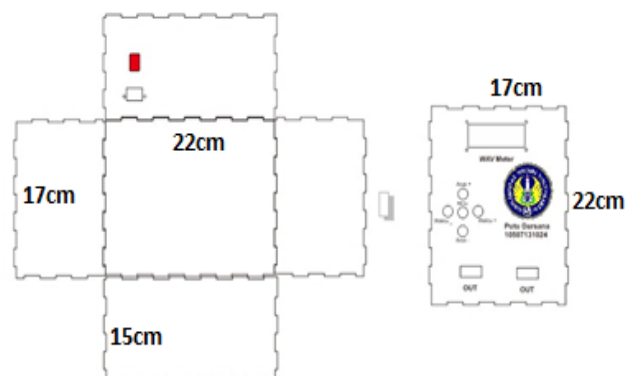
Lebar : 17 cm

Tinggi : 15cm

2. Pembuatan *box*

Box dibuat dari bahan *acrylic* dengan ketebalan 3mm. *Box* ini didalamnya terdapat 6 buah rangkaian yaitu rangkaian catu daya, sistem minimum, sensor tegangan, penguat instrumentasi, pendeteksi beda fasa, *relay* dan *buzzer*. serta 1 buah sensor arus *Split Current Transformer CT30*.

Berikut ini adalah gambar desain *box* yang menggunakan bahan *acrylic* :



Gambar 29. Desain *Box*

E. Spesifikasi Alat

Sistem Proteksi Daya Listrik ini dapat mempunyai spesifikasi sebagai berikut:

1. Tegangan kerja alat yaitu 220 VAC 50Hz
2. Terdapat *push button* yang berfungsi tombol menu dimana terdapat 3 menu yaitu menu batas arus, menu tampilan hasil pengukuran, dan menu waktu proteksi.
3. Menggunakan LCD 16 x 2 untuk menampilkan hasil pengukuran arus (I), tegangan (V), *cosphi* (FP) dan daya (P).
4. Batas ukur arus hingga 2 Ampere.
5. Batas waktu proteksi minimal 1 detik dan maksimal 60 detik.

F. Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data seberapa besar ketelitian alat dalam mengukur berbagai beban dan seberapa cepat respon alat memproteksi beban yang terpasang. Pengujian dilakukan dengan beberapa tahap yang meliputi:

1. Uji Fungsional

Pengujian dilakukan dengan menguji setiap bagian yaitu pengujian pada rangkaian catu daya, pengujian sensor tegangan, pengujian sensor arus, pengujian rangkaian penguat instrumentasi, pengujian rangkaian pendeteksi beda fasa, pengujian sistem minimum, dan pengujian rangkaian *relay* dan *buzzer*. Pengujian ini dilakukan agar dapat mengetahui fungsi masing-masing blok apakah bekerja dengan baik atau tidak dan dapat

mengetahui kekurangan yang ada sebelum semua rangkian dirakit ke dalam *box*.

2. Uji Keseluruh Sistem

Pengujian dilakukan dengan menguji alat yang sudah jadi berdasarkan fungsinya. Pengujian dilakukan dengan membandingkan hasil pembacaan dengan alat ukur Wattmeter, Multimeter, Cosphimeter, dan Clampmeter yang dianggap standar. Beban yang digunakan antara lain adalah menggunakan lampu bohlam, *ballast* lampu neon, strika, dan bor listrik AC. Hasil pengujian akan disajikan pada pembahasan di BAB IV.

G. Pengoperasian Alat

Alat dapat dioperasikan dengan langkah sebagai berikut:

1. Pastikan kabel power telah terhubung dengan tegangan AC 220 V.
2. Tekan tombol *Power* untuk menghidupkan Sistem Proteksi Daya Listrik.
3. Tunggu hingga Sistem Proteksi Daya Listrik dalam keadaan siap.
4. Tekan tombol menu hingga tampil menu pengaturan batas arus kemudian atur batas arus yang diinginkan selanjutnya tekan tombol menu hingga tampil menu waktu proteksi kemudian atur waktu proteksi yang diinginkan. Selanjutnya tekan tombol menu lagi hingga tampil hasil pengukuran.
5. Pasang beban di *jack AC Outlet* pada Sistem Proteksi Daya Listrik
6. Hidupkan beban.

H. Jadwal Pelaksanaan

Jadwal pelaksanaan pembuatan proyek akhir ini dimulai dari bulan desember 2013 sampai bulan januari 2015 dengan rincian pelaksanaan kegiatan seperti tabel di berikut ini:

Tabel 8. Jadwal pelaksanaan

N O	KEGIATAN	BULAN KE Tahun 2013- 2015					
		12/ 13	8/ 14	9/ 14	11/ 14	12/ 14	2/15
1.	Pengajuan proposal						
2.	Perancangan						
3.	Pembuatan dan pengujian per blok						
4.	Pengujian sistem dan pengumpulan data						
5.	Penyusunan laporan						
6.	Ujian Akhir						
7.	Revisi dan Penyerahan						

I. Daftar Komponen dan Biaya Pembuatan

Berikut ini adalah daftar komponen beserta rincian biaya yang digunakan untuk membuat Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega16 dengan sensor Arus.

Tabel 9. Daftar Komponen

No	Komponen	Nilai komponen	Jumlah	Harga
1	LCD 16 x 2		1	Rp 40.000
2	PCB	12 X 15 cm	3	Rp 18.000
3	<i>Relay</i>	SPDT 12V 10A/250V	2	Rp 12.000
4	<i>Buzzer</i>	12 - 24 VDC	1	Rp 15.000
5	Soket sisir	40 pin	2	Rp 6.000
6	ATmega 16		1	Rp 50.000
7	Sensor Arus AC	AC 75A	1	Rp 130.000
8	<i>Push Button</i>		5	Rp 15.000
9	Kabel pelangi 10 pin	3 meter	3	Rp 4.500
10	Kabel <i>power</i>	1 meter	1	Rp 8.000
11	Led	5 mm	3	Rp 1500
12	Soket 2 pin	2 pin	4	Rp 8.000
13	Trafo 1 A	CT 1A	1	Rp 22.000
14	Trafo 500 mA	Non CT 500mA	1	Rp 15.000
15	<i>Capasitor</i>	22 nF non polar	2	Rp 500
		22 00uF Polar	2	Rp. 6000
		6800 Uf Polar	1	Rp. 8000
		220 uF polar	1	Rp 800
16	Resistor	1 K Ohm ¼ watt	10	Rp 500
		10 K Ohm ¼ watt	10	Rp 500
		100 K ohm ¼ watt	2	Rp 100
		47 K ohm ¼ watt	4	Rp 200
17	<i>Crystal</i>	12 MHz	1	Rp 3000
18	Dioda	In 4001	10	Rp 5.000
19	Transistor NPN, PNP	TIP41,TIP42	6	RP. 9000
20	<i>BOX acrylic</i> , dll			Rp 85.0000
Jumlah				Rp.460.600

BAB IV

PROSES, HASIL, PENGUJIAN, DAN PEMBAHASAN

A. Proses Pembuatan

Proses pembuatan proyek akhir Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega16 dengan Sensor Arus terdiri dari beberapa langkah yaitu, merealisasikan program berdasarkan algoritma yang telah dibuat, kemudian merealisasikan *hardware* sesuai dengan rancangan.

1. Realisasi Program

Pembuatan program menggunakan salah satu *software compiler* bahasa C yaitu *CodeVision AVR*. Langkah pembuatan program dimulai dengan membuat *new project* kemudian mengatur parameter pada *Automatic Program Generator CodeVision AVR*, kemudian menuliskan programnya dan meng*compile* program tersebut yang akan menghasilkan berbagai jenis file antara lain .C, .hex, dan .coff. File *hex* inilah yang kemudian akan di *load* ke dalam sistem mikrokontroler.

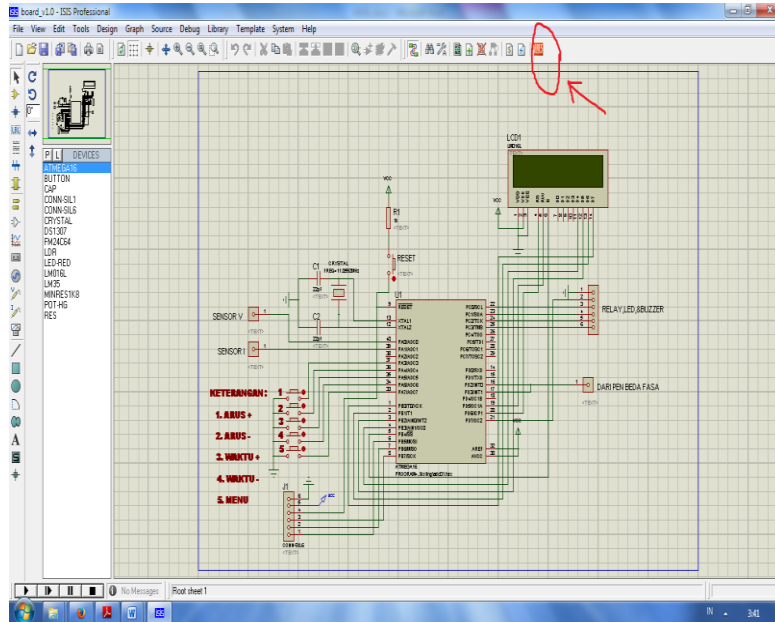
2. Realisasi Hardware

Dari rancangan *hardware* kemudian direalisasikan menjadi bentuk alat dengan beberapa tahapan berikut:

a) Membuat desain rangkaian dan jalur PCB

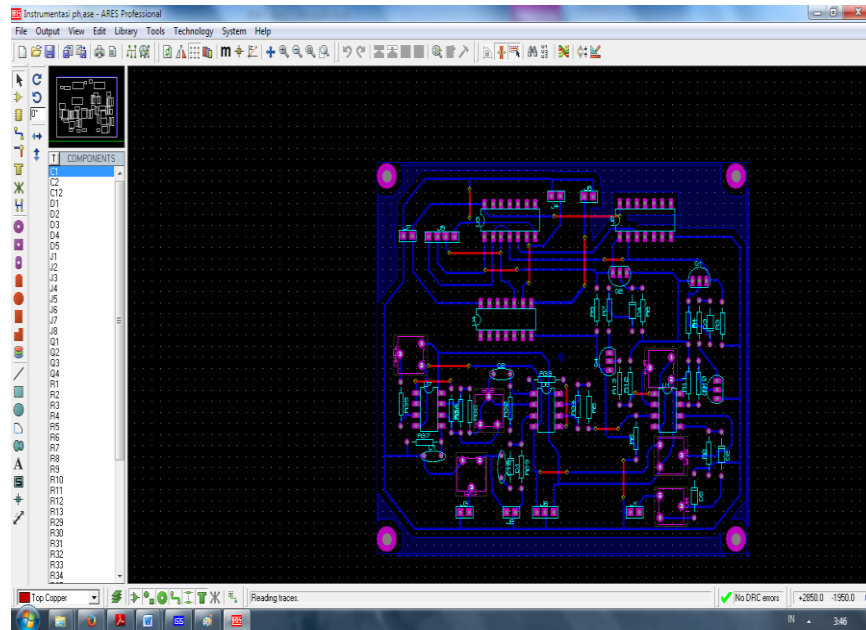
Dalam proses pembuatan desain rangkaian dan jalur PCB penulis menggunakan bantuan *software* ISIS Proteus versi 7.10 SP 2. Proses

pembuatan desain dengan menggunakan *software* ISIS Proteus dilakukan berulang sampai dengan semua bagian selesai.



Gambar 30. Ilustrasi pembuatan rangkaian pada ISIS

Gambar diatas menjelaskan proses perangkaian komponen sistem minimum yang dibutuhkan sesuai dengan rancangan, setelah proses penyambungan selesai kemudian langkah selanjutnya adalah meng"export" ke ARES dengan tombol yang ditandai lingkaran merah diatas. Pada tahap ini program akan meminta menyimpannya terlebih dahulu dengan ekstensi .DSN dan kemudian baru masuk ke ARES.



Gambar 31. Ilustrasi pembuatan PCB pada ARES

Setelah rangkaian di *export* ke ARES, langkah selanjutnya yaitu dengan mengatur tata letak komponen sesuai dengan keinginan, proses ini sangat penting karena akan mempengaruhi rumit tidaknya jalur yang akan dibuat. Setelah tata letak komponen sudah sesuai, langkah selanjutnya yaitu proses *routing*. Yaitu penyambungan jalur dari kaki setiap komponen. Proses ini dapat dilakukan dengan cara manual maupun dengan cara *automatic routing* yang tersedia pada program ARES. Setelah semua jalur telah terhubung dengan baik, kemudian langkah selanjutnya yaitu mencetak jalur tadipada kertas *glossy* menggunakan printer laser.

b) Proses penyablonan pada PCB polos

Setelah jalur PCB telah dicetak menggunakan printer laser, langkah selanjutnya yaitu menyablon dengan cara menempelkan bagian

yang terdapat tinta jalur rangkaian kertas glosy pada PCB polos yang sudah dibersihkan menggunakan sabun. Selanjutnya panaskan setrika, setelah panas setrika dirasa cukup kemudian gosokkan setrika perlahan-lahan sampai merata dan berulang ulang kurang lebih selama dua menit. Kemudian bersihkan kertas glosy pada PCB yang masih menempel hingga bersih sehingga hanya menyisakan jalur yang menempel pada PCB.



Gambar 32. Proses penyablonan menggunakan setrika pada papan PCB

c) Proses pelarutan

Setelah Proses penyablonan selesai selanjutnya adalah proses pelarutan PCB menggunakan pelarut yaitu cairan *feri clorida* ($FeCl_3$), dengan cara mencampur air dan *feri clorida* secukupnya pada wadah plastik kemudian goyang goyangkan cairan *feri clorida* dengan tujuannya adalah agar bagian yang tidak tertutup oleh tinta dapat hilang dan hanya tersisa jalur rangkaiannya. selanjutnya yaitu membersihkan

sisa-sisa tinta dengan *thinner* dan selanjutnya dicuci dengan air sabun hingga bersih.



Gambar 33. Proses pelarutan PCB dengan larutan *feri clorida*

d) Proses pengeboran

Setelah PCB sudah dilarutkan dan membentuk jalur rangkaian yang diinginkan, langkah selanjutnya yaitu mengebor setiap lubang tempat pemasangan komponen.



Gambar 34. Proses pengeboran PCB

e) Proses pemasangan komponen dan penyolderan

Setelah PCB tersebut berlubang kemudian memasang komponen komponen yang langsung disolder dengan urutan komponen pasif terlebih dahulu seperti resistor, kapasitor, baru kemudian komponen aktif seperti IC, transistor, dan yang lainnya.

f) Proses perakitan rangkaian pada *Box*

Box yang digunakan yaitu box transparan yang terbuat dari bahan *acrylic*, dalam proses pembuatan *box* penulis hanya membuat desainnya saja namun dalam proses pemotongan menggunakan bantuan jasa pemotongan *acrylic* untuk kemudian merakitnya menjadi box dengan lem. Selanjutnya yaitu melubangi *box* sesuai ukuran tempat lubang baut baik untuk memasang PCB, kaki *box* maupun untuk keperluan lain. Setelah dilubangi, kemudian memasang bagian-bagian yang diperlukan serta proses penyambungan kabel sehingga menjadi suatu alat utuh.

B. Hasil

1. Hasil Perakitan Alat

a) Hasil Rangkaian Catu Daya



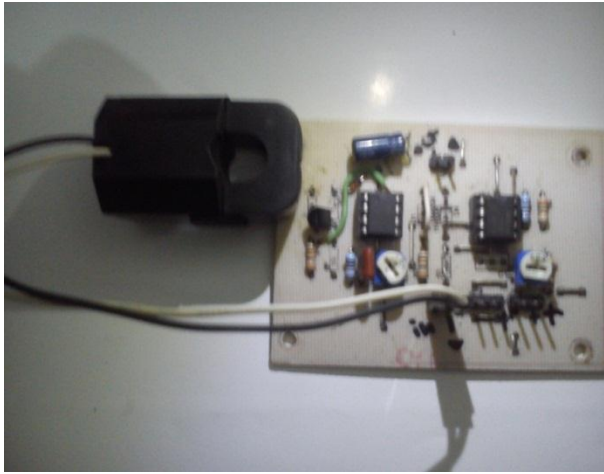
Gambar .35 Hasil Rangkaian Catu Daya

b) Hasil Rangkaian Sensor Tegangan



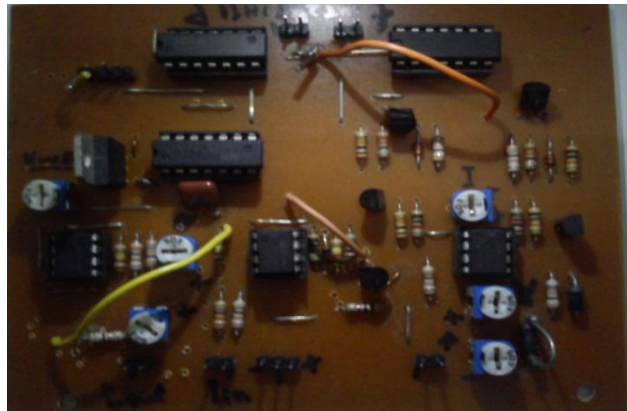
Gambar. 36 Hasil Rangkaian Sensor Tegangan

c) Hasil Rangkaian Penguat Instrumentasi



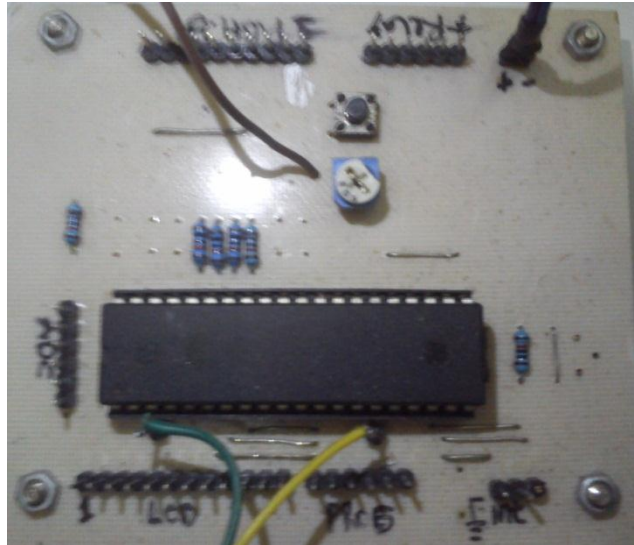
Gambar. 37 Hasil Rangkaian Penguat Instrumentasi

d) Hasil Rangkaian Pedeteksi Beda Fasa



Gambar. 38 Hasil Rangkaian Pedeteksi Beda Fasa

e) Hasil Rangkaian Sistem Minimum ATmega 16



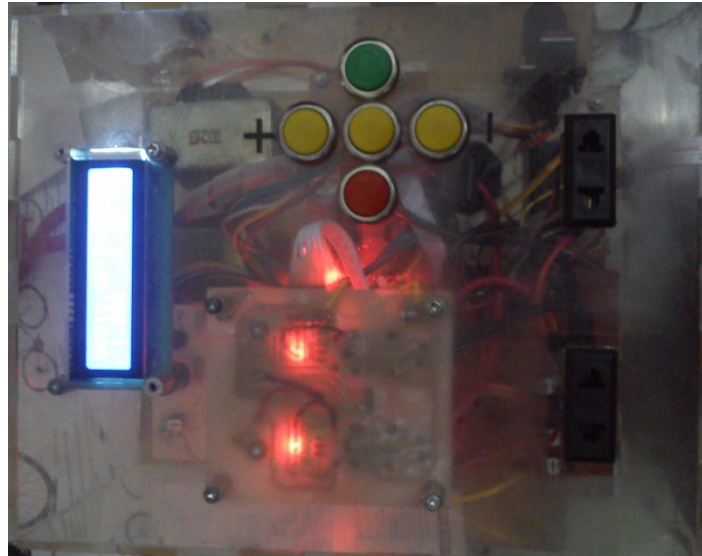
Gambar. 39 Hasil Rangkaian Sistem Minimum ATmega 16

f) Hasil Rangkaian *Driver Relay* dan *Buzzer*



Gambar. 40 Hasil Rangkaian *Driver Relay* dan *Buzzer*

g) Hasil Keseluruhan Alat



Gambar. 41 Hasil keseluruhan Alat

C. Pengujian

Pengujian Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus ini bertujuan untuk melihat apakah program yang telah dibuat sudah sesuai dengan rancangan. Pengujian ini dilakukan pada setiap blok rangkaian apakah sudah bekerja sebelum semua blok rangkaian disatukan sehingga alat dapat bekerja secara keseluruhan. Pengujian tersebut meliputi:

1. Pengujian Setiap Bagian

a) Pengujian Catu Daya

Catu daya yang dibutuhkan pada rangkaian proyek akhir ini adalah catu daya simetris (+ 8V dan – 8 V) digunakan untuk mensupply rangkaian penguat instrumentasi sensor arus dan rangkaian pedeteksi beda fasa.

Catu daya tunggal 5 V digunakan untuk rangkaian sistem minimum ATmega 16. Untuk catu daya tunggal 12 V digunakan untuk mensupply rangkaian *driver relay* dan *buzzer*. Untuk mengetahui menegetahui hasil pengujiannya dapat dilihat pada tabel berikut ini:

Tabel 10. Hasil pengujian Rangkaian Catu Daya

No	Bagian	Output Komponen	Tegangan Output	
			Tanpa Beban	Beban Penuh
1	AC Input Trafo	Konektor	221 VAC	221 VAC
2	AC Output Trafo	Konektor	12 VAC	12 VAC
3	Catu Daya simetris ± 8	TIP 41	+ 7,9 VDC	+ 7,9 VDC
		TIP 42	- 7,97 VDC	- 7,97 VDC
4	Catu Daya tunggal 5 V	LM 7805	4,78 VDC	4,75 VDC
5	Catu Daya tunggal 12 V	LM 7812	12.6 VDC	12.5 VDC

b) Pengujian blok rangkaian sensor tegangan

Pengujian blok rangkaian sensor tegangan dilakukan untuk mengetahui apakah tegangan yang diberikan yaitu tegangan DC dengan rentang 0-5V ke *PINC.0* yaitu salah satu *pin ADC* pada mikrokontroler ATmega 16. Hasilnya adalah seperti ditampilkan pada tabel 11:

Tabel 11. Hasil pengujian Rangkaian Sensor Tegangan

Input ADC (Volt)	Hasil pengukuran (V)
0	0
1	50,5
2	101,1
3	152,2
4	203,5
4.41	224,6
5	247,6

Tabel 12. Hasil perbandingan pengukuran Tegangan antara Multimeter digital dengan Sistem Proteksi Daya Listrik.



Input tegangan AC PLN	Multimeter digital (VAC)	Sistem Proteksi Daya Listrik (VAC)	Kondisi
220	224	224,6	berfungsi

Dari data diatas dapat dilihat bahwa rangkaian sensor tegangan yang dibuat sudah dapat mengukur tegangan dimana pada kondisi tegangan ADC 0 VAC, hasil pengukuran tegangan pada alat Sistem Proteksi Daya Listrik menunjukan 0 V juga. Sedangkan pada kondisi tegangan ADC 4.41 V, hasil pengukuran tegangan pada alat Sistem Proteksi Daya Listrik adalah 224.6 V. Untuk tegangan ADC 5 itu menunjukkan nilai tegangan 247.6 V

c) Pengujian sensor arus *Split Current Transformer CT301*

Pengujian ini untuk mengetahui apakah sensor arus dapat bekerja terhadap beban yang berikan serta untuk mengetahui bentuk sinyal *output* dari sensor arus *Split Current Transformer*

Tabel 13. Hasil pengujian Sensor Arus *Split Current Transformer*

Kondisi	Hasil pengukuran (V)	Bentuk sinyal
Tanpa Beban	0,2 V	
Beban 40 Watt	1,2 V	

Dari data pengukuran diatas dapat diketahui keluaran dari sensor arus dimana semakin besar arus pada beban maka semakin besar juga V_{out} dari sensor arus *Split Current Transformer CT30*.

d) Pengujian blok rangkaian Penguat Instrumentasi

Pengujian blok instrumentasi dilakukan untuk mengetahui nilai penguatan dari masukan yaitu sensor arus pengukuran dilakukan dengan menggunakan multimeter dengan melihat tegangan keluaran . Berikut ini adalah hasil pengukuran dari rangkaian penguat instrumentasi.



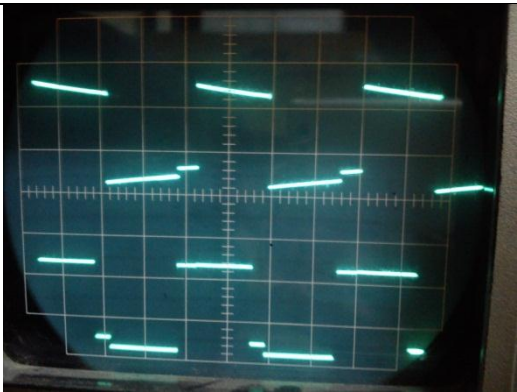
Tabel 14. Hasil Pengujian blok rangkaian Penguat Instrumentasi

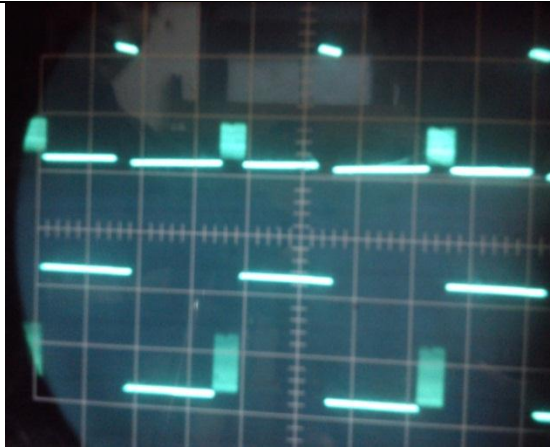
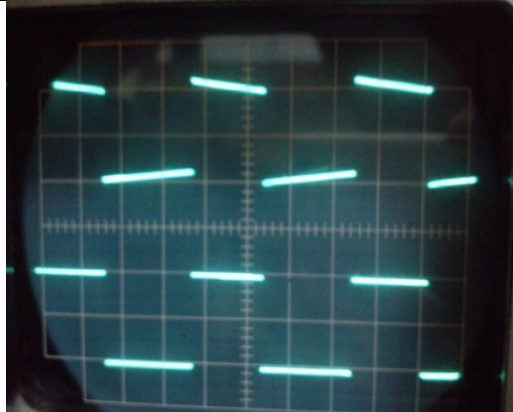
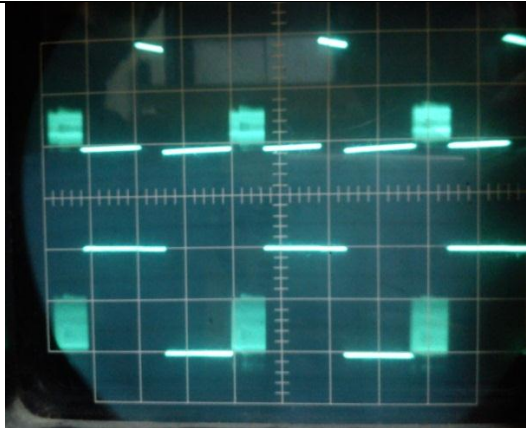
Nilai beban	Sinyal input	Sinyal output	Penguatan
30 watt	0,2 V	1 V	5 kali
65 watt	0,5 V	2,4 V	5 kali
100 watt	0,7 V	3,2 V	5 kali

e) Pengujian blok Rangkaian Pendeteksi Beda fasa

Pengujian blok rangkain beda fasa adalah untuk mengetahui perbandingan sinyal tegangan dengan sinyal arus pada rangkaian.

Tabel 15. Hasil pengujian Rangkaian Pendeteksi Beda Fasa

Jenis beban	Bentuk sinyal	Keterangan
-		Output sinyal sensor arus
-		Output sinyal tegangan dari Trafo
Solder 40 watt		Output perbandingan sinyal arus dan sinyal tegangan pada gerbang XOR

Solder 40 watt		Output perbandingan sinyal arus dan sinyal tegangan pada gerbang AND
Tanpa Beban		Sinyal atas output gerbang XOR Sinyal bawah output gerbang AND
Charger Laptop		Sinyal atas output gerbang XOR Sinyal bawah output gerbang AND

Dari data diatas dapat dilihat rangkaian pendeteksi beda fasa sudah berfungsi dimana nilai pengukuran berubah sesuai dengan beban yang dipasang.

f) Pengujian blok Rangkaian *Driver Relay dan Buzzer*

Pengujian blok *relay* dan *buzzer* dilakukan untuk mengetahui apakah *relay* dan *buzzer* dapat berfungsi dimana saat nilai batas arus ditentukan apakah *relay* akan aktif sehingga akan memutus arus dan *buzzer* akan berbunyi sebagai alarm.

Tabel 16. Hasil pengujian rangkaian *Driver Relay dan Buzzer*

Beban	Nilai batas arus yang ditentukan	Nilai pengukuran arus Sistem Proteksi Daya Listrik	Kondisi <i>Relay</i>	Kondisi <i>Buzzer</i>	Keterangan
Kipas angin	0,15	0,13	Non aktif	Tidak berbunyi	Berfungsi
<i>Chager</i> Laptop	0,22	0,25	Aktif	Berbunyi	Berfungsi
Bolham 40 W	0,17	0,17	Aktif	Berbunyi	Berfungsi
Solder	0,10	0,11	Aktif	Berbunyi	Berfungsi

Dari data diatas blok rangkaian *driver relay* dan *buzzer* sudah berfungsi sesuai dengan yang diinginkan dimana saat nilai batas arus beban yang ditentukan belum melebihi batas arus yang ditentukan maka *relay* dan *buzzer* tidak aktif.

Namun saat nilai arus beban melebihi nilai batas yang ditentukan *relay* langsung aktif untuk memutus arus beban dan *buzzer* berbunyi.

g) Pengujian Tampilan LCD

Pengujian LCD dilakukan untuk mengetahui apakah LCD menampilkan karakter sesuai dengan program yang dibuat kemudian di download ke mikrokontroler ATmega 16. Pengujian dilakukan dengan menggunakan program simulasi *proteus 7.10*. program yang akan di download pada ATmega 16 dan menampilkan pada LCD adalah sebagai berikut:

```

Void tampilkan_tegangan()
{
    temp=read_adc(0);
    vin=(float)temp*0.0055*44;
    ftoa(vin,1,tempat);
    lcd_gotoxy(0,0);
    lcd_putsf("V=");
    lcd_gotoxy(2,0);
    lcd_puts(tempat);
    lcd_gotoxy(7,0);
    lcd_putsf("V");
}

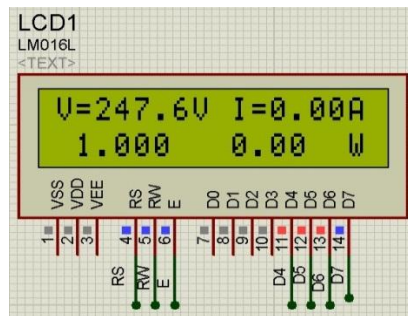
void tampilkan_arus()
{
    iin=(float)temp3*0.00067647;
    //batas ukur 2 A
    arus2=iin * 100;
    ftoa(iin,2,tempat);
    lcd_gotoxy(9,0);
    lcd_putsf("I=");
    lcd_gotoxy(11,0);
    lcd_puts(tempat);
    lcd_gotoxy(15,0);
    lcd_putsf("A");
}

void tampilkan_cos()
{
    //SP=0.0003846;
    ftoa(hsl_cospi,3,tempat);
    lcd_gotoxy(1,1);
    lcd_puts(tempat);
    lcd_gotoxy(0,1);
    if(data>0)
    {
        if(geser_phase==2){lcd_putsf("-");}
        if(geser_phase==3){lcd_putsf("+");}
        if(data==0){lcd_putsf("");}
    }

    delay_ms(1);
    data=0;
}

void tampilkan_watt()
{
    watt=vin*iin*hsl_cospi;
    ftoa(watt,2,tempat);
    lcd_gotoxy(9,1);
    lcd_puts(tempat);
    lcd_gotoxy(15,1);
    lcd_putsf("W");
    delay_ms(1);
    freq=0;
    data=0;
    conversi(0,0);
}

```



Gambar.42 Tampilan LCD pada simulasi

h) Pengujian *Push Button*

Pengujian *push button* dilakukan apakah fungsi input dari alat yang berguna untuk menjadi inputan pengaturan sudah berfungsi dimana saat *push button* ditekan maka perubahan akan terlihat pada LCD. Berikut ini adalah cuplikan program *push button*:

```
void menu()
{if(menu1==0)
  {display++;
  if(display==5){display=0;}
  delay_ms(100);
}
if(display==0)
{tampilkan_tegangan();
tampilkan_arus();
conversi(data,freq);
tampilkan_cos();
tampilkan_watt();}
else if(display==1)
{lcd_gotoxy(1,0);
lcd_putsf("SET ARUS");
arus1=(float)arus*0.01;
arus3=arus1 * 100;
ftoa(arus1,2,tempat);
lcd_gotoxy(3,1);
lcd_puts(tempat);
lcd_gotoxy(0,1);
lcd_putsf("PF=");
if(lplus==0)
{arus++;
if(arus>=1000){arus=0;}
delay_ms(100);
}
else if(lmin==0)
{arus--;
if(arus<0){arus=1000;}
delay_ms(100);
}
}
if(display==4)
{freq_pln();
tampilkan_arus();
phase_beban();
cosq_beban();
}
if(display==3)
delay_ms(100);
}
}
else if(display==2)
{lcd_gotoxy(1,0);
lcd_putsf("SET      TIMER
BEBAN");
ftoa(jeda,2,tempat);
lcd_gotoxy(3,1);
lcd_puts(tempat);
lcd_gotoxy(0,1);
lcd_putsf("PF=");
if(Tplus==0)
{jeda++;
if(jeda>=59){jeda=0;}
delay_ms(100);
}
else if(Tmin==0)
{jeda--;
if(jeda<0){jeda=59;}
delay_ms(100);
}
}
}
```



```

{lcd_gotoxy(1,0);
lcd_putsf("SET TIMER CEK");
ftoa(cek,2,tempat);
lcd_gotoxy(3,1);
lcd_puts(tempat);
lcd_gotoxy(0,1);
lcd_putsf("PF=");

```

```

if(Tplus==0)
{cek++;
if(cek>=59){cek=0;}
delay_ms(100);
}

```

2. Pengujian Keseluruhan

Pengujian keseluruhan dilakukan untuk mengetahui ketepatan alat dalam pengukuran dan mengetahui apakah sistem proteksi dan alarmnya sudah berfungsi. Pengujian dilakukan dengan membandingkan Sistem Proteksi Daya Listrik dengan alat ukur lain diantaranya Wattmeter analog Merk Takemoto PD310, Multimeter digital Krisbow, cosphi meter Yokogawa Type 2039, dan Clamp meter DT3266L.

a) Pengujian Beban Resistif

Pengujian dengan beban resistif menggunakan Lampu Bohlam Phillips. Hasilnya adalah sebagai berikut :

Tabel 17. Pengujian beban Resistif dengan Lampu Bohlam

Beban		Lampu Bohlam Phillips		
Daya nominal (watt)		40	60	100
Sistem Proteksi Daya Listrik	Watt	40	60	100
	Ampere	30,45	48,27	83,43
	Volt	0,16	0,25	0,43
	Cosphi	213	213	213
Alat ukur pembanding	Wattmeter Yokogawa (W)	-0,89	-0,90	-0,91
	Voltmeter Krisbow (V)	38	56	96
	Clampmeter DT3266L (A)	215	215	215
	Cosphimeter Takemoto	0,17	0,26	0,44

Keterangan: Tanda minus (–) menandakan kondisi LAGGING

Secara teori, perhitungan bebannya adalah sebagai berikut:

Sebagai contoh, bila:

Nilai hambatan bolham =R

$R = 290 \, \Omega$

$P = V * I * \cos \phi$

$V = 220 * I * 1$

$I = V/R = 220/290 = 0,758$

$P = 220 * 0,758 * 1 = 166,76 \, W$

$\cos \phi = 1$ karena beban resistif murni

b) Pengujian dengan beban Kapasitif

Pengujian dengan beban kapastif ini lampu bolham 40watt yang dipasang seri dengan Kapasitor. Hasilnya adalah seperti tertampil pada tabel berikut

Tabel 18. Pengujian beban Kapasitif

Beban		Lampu Bolham 40 watt + Kapasitor		
Nilai nominal Kapasitor		7,5 uF	15 uF	22,5 uF
Sistem Proteksi Daya Listrik	Watt	26,24	29,69	30,77
	Ampere	0,16	0,17	0,17
	Volt	213	213	213
	Cosphi	+ 0,77	+0,82	+0,85
Alat ukur pembanding	Wattmeter Yokogawa (W)	38	38	38
	Voltmeter Krisbow (V)	215	215	215
	Clampmeter DT3266L (A)	0,17	0,17	0,17
	Cosphimeter Takemoto	Lead 0,90	Lead 0,93	Lead 0,96

Keterangan: Tanda plus (+) menandakan kondisi LEADING

Secara perhitungan besarnya beban dapat dihitung menggunakan persamaan apabila :

$$R = 350 \, \Omega$$

$$C = 7,5 \, \mu\text{F} \times 3 = 22,5 \, \mu\text{F}$$

$$V = 220 \, \text{V}$$

$$P = V \cdot I \cdot \cos \phi$$

$$I = V / Z$$

$$Z = \sqrt{R^2 + X_c^2}$$

$$X_c = \frac{1}{2\pi \cdot f \cdot C}$$

$$X_c = \frac{1}{2\pi \cdot 50 \cdot 22,5 \cdot 10^{-6}}$$

$$X_c = 141,5 \, \Omega$$

$$Z = \sqrt{R^2 + X_c^2}$$

$$Z = \sqrt{350^2 + 141,5^2}$$

$$Z = 377,5 \, \Omega$$

$$\cos \phi = R / Z$$

$$\cos \phi = 350 / 377,5 = 0,93$$

$$I = V / Z$$

$$I = 220 / 377,5 = 0,58 \, \text{A}$$

$$P = V \cdot I \cdot \cos \phi$$

$$P = 220 \cdot 0,58 \cdot 0,93 = 118,67 \, \text{W}$$

c) Pengujian beban induktif

Pengujian ini menggunakan beban Ballast Lampu TL

Tabel 19. Pengujian menggunakan beban Induktif

Beban		Ballast Lampu TL		
Daya nominal (watt)		10	18	20
Sistem Proteksi Daya Listrik	Watt	12,45	16,77	21,23
	Ampere	0,06	0,10	0,15
	Volt	213	213	213
	Cosphi	- 0,82	- 0,73	- 0,67
Alat ukur pembanding	Wattmeter Yokogawa (W)	16	26	30
	Voltmeter Krisbow (V)	215	215	215
	Clampmeter DT3266L (A)	0,13	0,23	0,27
	Cosphimeter Takemoto	Lag 0,6	Lag 0,5	Lag 0,5

Keterangan : Tanda minus (-) menandakan kondisi LAGGING

D. Pembahasan

Berdasarkan hasil pengujian tiap bagian dan pengujian secara keseluruhan, maka dapat diuraikan beberapa permasalahan yang berhubungan dengan kinerja alat dalam pembahasan berikut:

1. Hardware

a) Rangkaian catu Daya

Rangkaian catu daya ini tersusun atas dua blok yaitu rangkaian catu daya tunggal dan rangkaian catu daya simetris. Dalam proyek ini sumber catu daya menggunakan catu daya tunggal sebesar 12 Vdc yang digunakan untuk rangkaian driver *relay* dan *buzzer*, dan catu daya tunggal sebesar 5 Vdc yang digunakan untuk rangkaian sistem minimum mikrokontroler ATmega 16. Untuk bagian lain menggunakan catu daya simetris (+ 8 Vdc dan - 8Vdc) yang digunakan pada rangkaian penguat instrumentasi dan rangkaian pendeteksi beda fasa.

b) Rangkaian Sensor Tegangan

Rangkaian sensor tegangan ini berfungsi untuk mengukur tegangan pada beban kemudian ditampilkan pada LCD. Cara kerjanya yaitu PIN ADC.0 pada mikrokontroler membaca tegangan DC yang keluar dari rangkaian ini untuk dikonversikan dan ditampilkan melalui LCD. Proses perubahan dan penurunan tegangan dari AC ke DC menggunakan trafo *step-down* kemudian disearahkan menggunakan dioda dan kapasitor, sama seperti rangkaian *power supply*. Dari hasil uji keseluruhan rangkaian ini sudah dapat berfungsi dengan baik dan cukup akurat, perbedaan disebabkan

karena tegangan AC juga terkadang mengalami ketidakstabilan tetapi masih dalam batas normal.

c) Sensor Arus *Split Current Transformer CT301* dan Rangkaian Penguat Instrumentasi

Pada proyek ini *split current transformer CT301* digunakan sebagai sensor arus dengan batas ukur maksimal dari sensor tipe ini sebesar 75A dengan bentuk sinyal output yaitu AC. Namun pada proyek ini batas ukur maksimal yang digunakan adalah sebesar 2A karena pada saat beban 2 ampere output dari sensor sudah maksimal ini disebabkan oleh sensitivitas arus terhadap beban sangat tinggi. Karena sinyal output sensor yang masih berbentuk sinyal AC maka perlu dikonversi kedalam bentuk sinyal DC dan akan dikuatkan dengan rangkaian penguat instrumentasi agar dapat dibaca oleh mikrokontroler ATmega 16. Pada rangkaian penguat instrumentasi ini terdiri dari dua buah IC Op – Amp LM358 yang disusun sesuai dengan IC INA 122. Pada proyek ini, hanya menggunakan batas ukur sebesar 2 A, tetapi pada dasarnya bisa diubah menjadi batas ukur lain dengan merubah penguatannya terlebih dahulu dengan cara menambah rangkaian *range selektor* sama halnya pada multimeter yang memiliki *range selektor*.

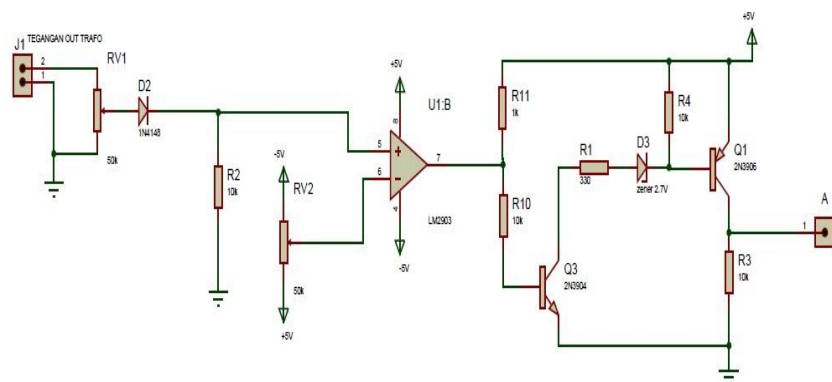
d) Rangkaian Pendeteksi Beda Fasa

Rangkaian pendeteksi beda fasa terdiri atas rangkaian penguat *Op-Amp* dan gerbang logika.

Rangkaian penguat *Op-Amp* terbagi menjadi dua bagian yaitu penguat sinyal tegangan dan penguat sinyal arus (beban). berikut ini penjelasan dari masing-masing rangkaian tersebut

(1) Rangkaian penguat *Op - Amp* sinyal tegangan.

Rangkaian ini terdiri dari penguat komparator menggunakan IC tipe LM2903 sinyal yang didapat dari *output* salah satu fasa trafo (input non inverting) dibandingkan dengan V_{ref} dari input (-). Apabila $V_{in} (+) >$ dari $V_{ref} (-)$ maka *output* dari IC komparator akan berlogika satu. *Output* dari *Op-amp* tersebut diteruskan kerangkaian *buffer* kemudian diteruskan ke rangkaian gerbang logika untuk proses pengolahan sinyal. Berikut salah satu contoh perhitungan untuk mengetahui V_{out} dari penguat tersebut:



Gambar. 43 Rangkaian komparator sinyal tegangan

Missal : $V_{in} (+) = 2\text{Volt}$. Agar berlogika 1 maka $V_{in} (-) = V_{ref}$ diatur pada tegangan.

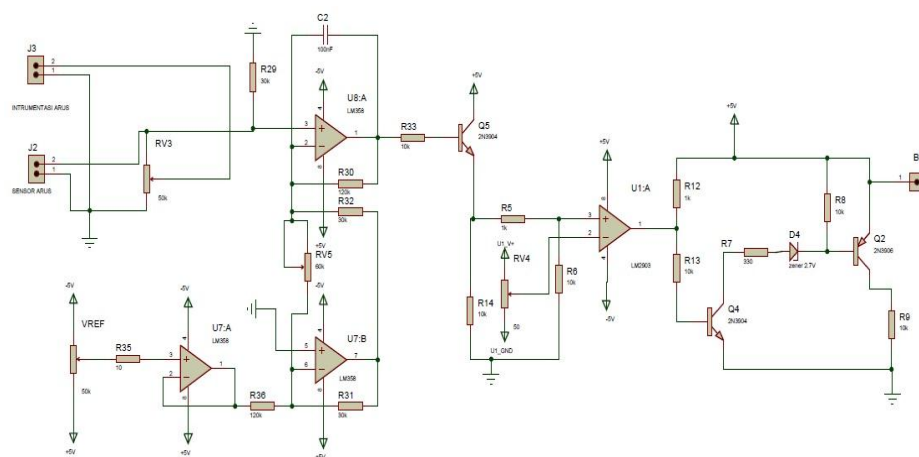
$$V_{out} = -V_1(V_{ref}) + V_2$$

$1\text{Volt} = V_1 + 2\text{Volt}$, maka $V_{in} (-) = V_{ref}$ diatur pada posisi tegangan -1Volt .

(2) Rangkaian penguat *Op – Amp* penguat sinyal arus

Rangkaian ini terdiri dari rangkaian penguat instrumentasi yang menggunakan IC tipe LM358 yang disusun menyerupai IC INA122 dan rangkaian komparator menggunakan IC tipe LM2903.

Berikut salah satu contoh perhitungan untuk mengetahui V_{out} dari rangkaian penguat sinyal arus :



Gambar. 44 Rangkaian penguat *Op-Amp* sinyal arus

Besarnya V_{out} penguat instrumentasi dapat dihitung dengan mencari besarnya resistor *gain*. Resistor *gain* pada

rangkaian penguat instrumentasi dapat dihitung dengan cara melihat tabel faktor *gain* pada IC seri INA122 yaitu sebesar 1026Ω. Sehingga bila dihitung dengan rumus dapat didapatkan sebagai berikut:

Misal : $V_{ref}=1\text{ Vdc}$

$$V_{in+} = 2\text{mVdc}$$

$$V_{in-} = 0\text{ Vdc}$$

$$R_G = 1026\ \Omega$$

$$G = 5 + \frac{200k}{R_G}$$

$$G = 199,93 \text{ atau } 200 \text{ kali}$$

$$V_{out} = V_{ref+} (V_{in+} - V_{in-}).(G)$$

$$V_{out} = 1 + (0,002-0) .(200) = 1.4\text{ Vdc}$$

Sedangkan untuk besarnya *output* dari rangkaian komparator sinyal arus dapat dihitung seperti berikut :

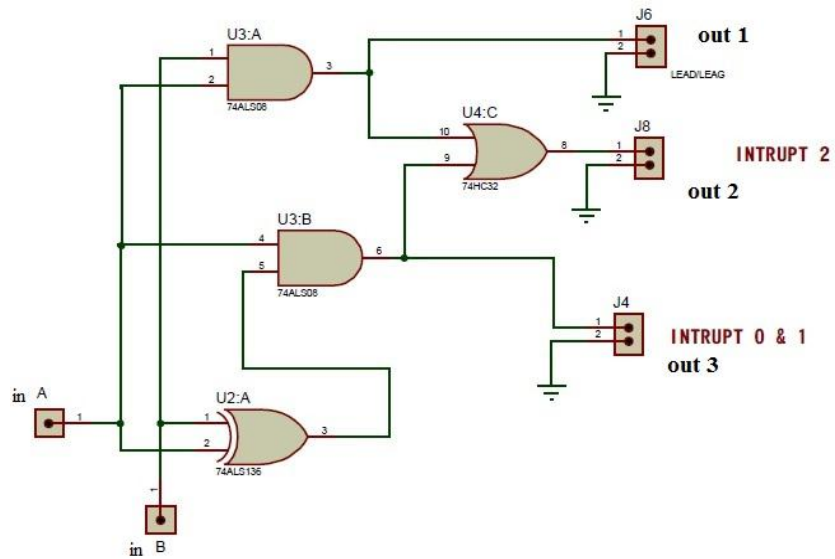
$V_{in (+)} = 1.4\text{ Volt}$. Agar berlogika 1 maka $V_{in (-)}$ V_{reff} diatur pada tegangan.

$$V_{out} = -V_1(V_{ref}) + V_2$$

$1\text{Volt} = V_1 + 1,4\text{Volt}$, maka $V_{in (-)}$ V_{reff} diatur pada posisi tegangan $-0,4\text{ Volt}$.

(3) Rangkaian gerbang logika

Rangkaian gerbang logika pada detektor beda fasa terdiri dari gerbang AND (7408), OR (74312) dan XOR (7486).



Gambar. 45 Rangkaian Gerbang Logika

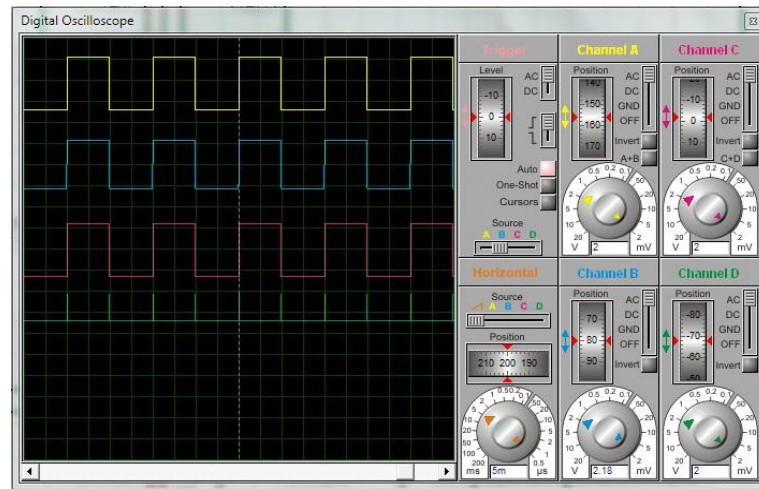
Dengan rumus output sebagai berikut :

$$\text{Output 1} = A.B$$

$$\text{Output 2} = (A(+)B).A+(A.B)$$

$$\text{Output 3} = (A(+)B).A$$

(a) *Output* beban Resistif



Gambar. 46 *Output* sinyal beban Resistif

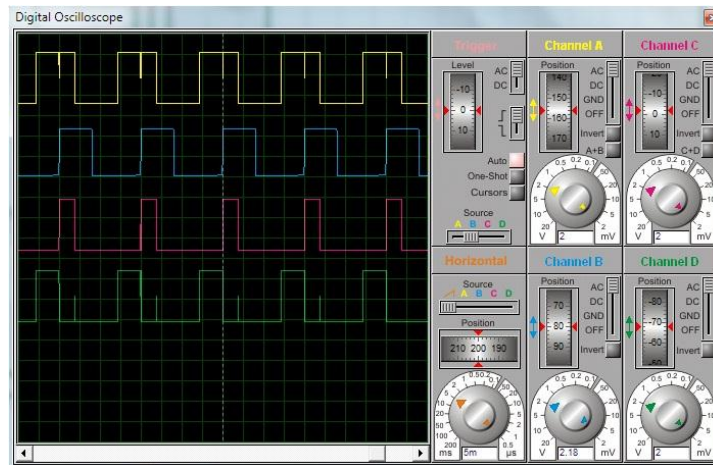
Tabel 20. Keluaran rangkaian gerbang logika pada saat beban resistif

Input A	Input B	Output 1	Output 2	Output 3	Ket
0	0	0	0	0	-
1	1	1	1	0	-

Pada gambar 46 *channel A* adalah output 2 sama dengan sinyal tegangan jala-jala PLN, *channel B* adalah sinyal beban (arus AC), *channel C* adalah sinyal yang digunakan pada program untuk menentukan posisi *leading* atau *leaging*, dan *channel D* adalah sinyal yang menghitung lebar pulsa beda fasa.

Sebagai contoh $\text{Output 3} = (A(+))B$.A jika A dan B = 1. Maka $\text{output 3} = 0$,1=0 dimana nilai ini akan diolah pada program sehingga didapat nilai beda fasa sebesar 1 karena beban resisitif.

(b) *Output* beban induktif 1000 Henry



Gambar. 47 *Output* sinyal beban Induktif

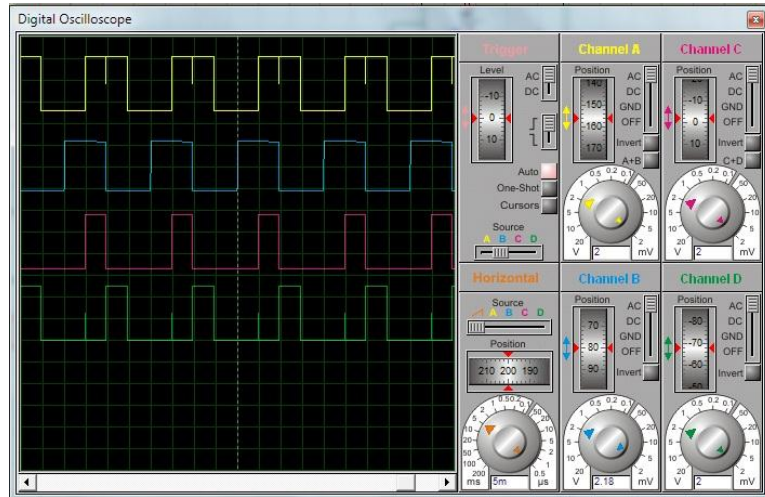
Tabel 21. Keluaran rangkaian gerbang logika pada saat beban induktif

Input A	Input B	Output 1	Output 2	Output 3	Ket
0	0	0	0	0	-
1	0	0	1	1	Penentu lebar pulsa $\cos \phi$
1	1	1	1	0	-
0	1	0	0	0	-

Pada Gambar. 47 merupakan bentuk sinyal beban Induktif

Dimana besarnya nilai beda fasa (*Chanel C*) didapat dari pengolahan pergeseran lebar pulsa antara sinyal tegangan (*chanel A*) dengan sinyal arus (*chanel B*). Pada gambar lebar sinyal *chanel A* mendahului sinyal *chanel B* sebesar 50%. Setelah diolah pada program maka ditampilkan pada LCD menjadi $\cos \phi = -0,5$.

(c) *Output* beban Kapasitif 100nF



Gambar. 48 Output sinyal beban Kapasitif

Tabel 22 . Keluaran rangkaian gerbang logika pada saat beban kapasitif

Input A	Input B	Output 1	Output 2	Output 3	Ket
0	0	0	0	0	-
0	1	0	0	0	
1	1	1	1	0	-
1	0	0	0	1	Penentu lebar pulsa $\cos \phi$

Pada gambar 48 terlihat bentuk sinyal tegangan (*channel A*) tertinggal oleh oleh sinyal arus (*channel B*). Sebesar 30 % Setelah diolah pada program maka ditampilkan pada LCD menjadi $\cos \phi = + 0,3$.

e) Sistem Minimum Mikrokontroler ATmega 16

Sistem Minimum ATmega16 berfungsi sebagai unit pemroses secara keseluruhan, baik sebagai pemroses perhitungan data maupun sebagai penampil data yang akan ditampilkan pada LCD. Dalam proyek ini, pada sistem minimum terdapat beberapa fitur yang digunakan yaitu ADC, Interrupt, dan Timer.

(1) ADC (*Analog digital converter*)

Analog digital converter (ADC) pada ATmega 16 digunakan untuk mengkonversi besaran tegangan V_{out} rangkaian sensor tegangan dan V_{out} sensor arus yang sudah dikuatkan menggunakan rangkaian penguat Instrumentasi kedalam satuan data digital desimal.

Port ADC yang digunakan dalam proyek akhir ini adalah *Pin ADC (0)* untuk rangkaian sensor tegangan dan *Pin ADC (1)* untuk pembacaan sensor arus dimana besarnya data digital yang digunakan sebesar 10 bit dengan frekuensi sampling 750.00KHz nilai tersebut merupakan nilai terbesar untuk lebar data frekuensi tertinggi sampling ADC. Hasil pengujian membutuhkan tegangan V_{ref} untuk mencapai nilai tertinggi yaitu sebesar 4,85 Vdc.

(2) Interrupt

Pada proyek ini *Interrupt* berfungsi sebagai pembaca sinyal dari rangkaian detektor beda fasa dan Pin yang digunakan yaitu *PIND.2*, *PIND.3* dan *PINB.2* dimana pada *PIND.2* dan *PIND.3*

dihubungkan secara paralel. *Pin* ini merupakan kanal *interrupt* eksternal pada ATmega 16.

Timer dalam perancangan program ini difungsikan untuk menghitung lebar pulsa sinyal frekuensi dari sumber PLN dan frekuensi beda fasa yang masuk dari *interrupt* rangkaian pendeteksi beda fasa. Penulis merancang timer sebesar 100 μ S dimana frekuensi sistem yang digunakan adalah 12 MHz dengan mode *Timer 16 bit* dengan *prescaler* 8. Perhitungannya adalah sebagai berikut :

$$TCNT = (FFFF + 1) - \frac{(T_{timer} \times fCLK)}{N}$$

$$TCNT = (10000h) - \frac{(0.0001 \times 12000000)}{8}$$

$$TCNT = 10000h - 150d$$

$$TCNT = 10000h - 96h$$

$$TCNT = FF6A$$

Sehingga didapat nilai *timer*

$$TCNTH = FF$$

$$TCNTL = 6A$$

(3) *Push Button*

Push button berfungsi untuk mengoperasikan *input* data pengaturan batas arus dan waktu proteksi kedalam mikrokontroler ATmega 16. Port A pada ATmega 16 digunakan untuk komunikasi *push button*. Berikut ini *Port* yang digunakan untuk komunikasi *push button*.

Tabel 23. Port I/O untuk operasi fungsi *Push Button*

No	Port A	Fungsi Pengaturan
1	Port A.3	Arus +
2	Port A.4	Arus -
3	Port A.5	Waktu +
4	Port A.6	Waktu -
5	Port A.7	Menu

(4) Tampilan LCD

LCD berfungsi untuk menampilkan hasil pengukuran tegangan, arus, faktor daya, dan daya serta berfungsi juga untuk menampilkan input data pengaturan batas arus dan waktu proteksi. Teknik antar muka yang digunakan untuk penampil LCD yaitu sebesar 4 bit. Pada tabel berikut ini adalah penjelasan konfigurasi LCD .

Tabel 24. Konfigurasi Pin LCD 16 x 2

Pin ATmega 16	Pin LCD M1632
VCC	3 (VCC)
GND	1 (GND)
PD.6	4 (RS)
PD.5	5 (RW)
PB.4	6 (E)
PB.3 - PB.0	11-14 (D4-D7)

f) *Rangkaian Driver Relay dan Buzzer*

Cara kerja rangkaian ini adalah dengan memberi logika 0 pada input driver relay maka relay akan aktif dikarekan posisi *relay* yang digunakan adalah *normaly close*. sedangkan untuk *driver buzzer* mendapat logika 1 agar bisa aktif. Rangkaian *driver relay* akan aktif apabila batas pengaturan arus pada beban melebihi data yang dimasukkan maka *relay* akan memutus arus dan *buzzer* berbunyi.

2. *Software*

Perangkat lunak dalam alat ini dibuat menggunakan bahasa pemrograman C, dengan memanfaatkan *compiler Code Vision AVR*. Dengan penggunaan *compiler CVAVR* inilah pemrograman dapat dilakukan secara cepat dan lebih sederhana. Bagian-bagian pemrograman meliputi:

a) **Definisi prosesor**

```
#include <mega16a.h>
```

Baris ini menyatakan bahwa chip yang digunakan adalah keluarga *AVR ATmega* dengan seri 16

b) **Penyertaan fungsi**

```
#include <delay.h>
```

Fungsi *header* bawaan dari *Code Vision AVR* yang harus di ikut sertakan dalam program jika kita akan menggunakan fungsi yang berkaitan dengan waktu.


```
#include <stdlib.h>
```

```
#include <alcd.h>
```

Fungsi *header* bawaan dari *Code Vision AVR* yang harus di ikut sertakan dalam program jika kita akan menggunakan fungsi yang berkaitan dengan instruksi untuk menggunakan LCD.

c) Definisi variable

Variable adalah pemberian type data pada suatu variable, hal ini dilakukan untuk menentukan tepatnya type data yang diberikan terhadap daya tampung. Berikut ini penulisan variable dalam program:

```
float vin,iin,watt,data,freq,hsl_cospi,SP,arus1;
int
tanda,tanda1,cv,display=0,timer_arus=0,timer_volt=0,arus_
_adc[20],volt_adc[20];
unsigned int
temp,temp1=0,temp2,arus2=0,arus3=0,array_arus=0,array_vo
lt=0;
long int usec,sec,usec1,sec1;
unsigned long int a,b,c,d,e,f,g;
char
tempat[33],arus_tanda=0,beban=0,geser_phase=0,factor=0;
eeprom signed int arus,jeda,cek;
```

Baris ini digunakan untuk mendeklarasikan variable yang digunakan dalam program utama.

d) Definisi Mode Timer1

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x6A;
// Place your code here
if(++usec1==10000)
{if(++sec1==60 &&
arus_tanda==1){sec1=0;}
usec1=0;
}
}
int
secv,opsi_waktu=0,timer_
opsi=0,u_timer_opsi=0;
// Timer1 overflow interrupt
service routine
```

```

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Place your code here
    TCNT1H=0xFF;
    TCNT1L=0x6A;
    // Place your code here
    if(tanda==1)
    {data=data+1;
    }
    if(++usec==10000)
    {if(++sec==60    &&
    timer_arus==1){sec=0;}
    if(++secv==6    &&
    timer_volt==1){secv=0;}

    usec=0;
    }
    if(tanda1==1)
    {freq=freq+1;
    }
    if(opsi_waktu==1)
    {if(++u_timer_opsi==10000)

    {if(++timer_opsi==60){time
    r_opsi=0;}
    u_timer_opsi=0;
    }
    }
}

```

Potongan program diatas merupakan bagian *sub rutin* yang difungsikan sebagai *timer*. Satu kali eksekusi dalam *sub rutin* ini dengan kecepatan 10^{-4} s. Setiap 10^4 kali eksekusi sama dengan 1 detiknya variabel *timer_opsi* dan *sec1*. Fungsi variabel *timer_opsi* digunakan sebagai batas waktu lamanya bunyi *buzzer* saat batas beban yang ditentukan sudah terlampaui. Variabel *sec1* digunakan sebagai lamanya jeda waktu *relay* memutus arus beban. Variabel *tanda* dan *data* digunakan untuk menghitung lebar pulsa beda fasa *data=data+1*. Variabel *tanda1* dan *freq* digunakan untuk menghitung lebar pulsa frekuensi PLN.

e) Definisi Port dan Mode ADC

```
#define ADC_VREF_TYPE 0x00
```

Kode program diatas digunakan untuk tegangan referensi yang digunakan adalah *pin Vreff* pada kaki ATmega16.

```
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
```

Baris kode program tersebut digunakan untuk menggabungkan kode hasil pemilihan *pin* tegangan referensi ADC dengan *pin* ADC yang digunakan, yaitu `read_adc(0)` untuk membaca sensor tegangan sedangkan `read_adc(1)` digunakan untuk membaca sensor arus. Hasil penggabungan dua variabel dengan perintah "Or" tersebut dimasukkan ke *register ADMUX* untuk melakukan konversi data.

f) Definisi Port dan Mode Interrupt

```

    interrupt [EXT_INT0] void
    ext_int0_isr(void)
    {
        // Place your code here

        if(PIND.2==1 && arus2>0 &&
PIND.1==1)
        {tanda=1;
        }

        //dataku=dataku + 1;

        if(PIND.1==1)
        {tanda1=1;
        }
    }

    // External Interrupt 1 service
    routine

    interrupt [EXT_INT1] void
    ext_int1_isr(void)
    {
        // Place your code here

        if(PIND.3==0 && arus2>0 &&
PIND.1==1
        &&
        factor==0){geser_phase=2;tanda=0
        ;}

        if(PIND.3==0 && arus2>0 &&
PIND.1==0
        &&
        factor==2){geser_phase=3;tanda=0
        ;}

        if(data>freq && PIND.1==1
        && arus2>0){factor=2;}

        if(data>freq && PIND.1==0
        && arus2>0){factor=0;}

        }

    interrupt [EXT_INT2] void
    ext_int2_isr(void)
    {
        // Place your code here

        if(PINB.2==0)

        {tanda1=0;
        }

        }

```

Baris ini program yang digunakan int 0, int 1, dan int 2. Pin int 0 dan 1 digabung untuk proses pembacaan beda fasa dengan pengaturan int 0 *rissing edge* apabila perubahan pulas 0 ke 1 maka int0 akan berintrupsi menghitung waktu lebar pulsa. Int 1 berada pada pengaturan *Falling Edge* dimana jika ada perubahan pulsan 1 ke 0 maka int 1 akan berintrupsi menghentikan proses *counter* lebar pulsa (beda fasa). Pin int 2 diatur pada posisi *rissing edge* apabila perubahan pulas 0 ke 1 maka int2 akan berintrupsi menghitung waktu lebar pulsa frekuensi tegangan PLN.

g) Definisi Sub Rutin

Sub rutin adalah program yang dibuat secara penggalan blok program untuk mempermudah didalam pemograman program utama. Ketika program utama membutuhkan code program pada sub rutin, maka program utama tinggal memanggil program sub rutin tersebut. Penggalan – penggalan pada sub rutin tersebut adalah sebagai berikut :

(1) Sub Rutin push button

```
void menu()
{if(menu1==0)
{display++;
```

Baris program ini digunakan untuk mendeteksi adanya perubahan data 0 pada setiap *pin push button*.

(2) Sub rutin Display

<pre>void menu() {if(menu1==0) {display++; if(display==5){display=0;} delay_ms(100);</pre>	<pre>} if(display==0) {tampilkan_tegangan(); tampilkan_arus(); conversi(data,freq);</pre>
--	---

```

    tampilkan_cos();
    tampilkan_watt();}
else if(display==1)
{lcd_gotoxy(1,0);
 lcd_putsf("SET ARUS");
 arus1=(float)arus*0.01;
 arus3=arus1 * 100;
 ftoa(arus1,2,tempat);
 lcd_gotoxy(3,1);
 lcd_puts(tempat);
 lcd_gotoxy(0,1);
 lcd_putsf("PF=");
 if(lplus==0)
 {arus++;
  if(arus>=1000){arus=0;}
  delay_ms(100);
 }
else if(lmin==0)
 {arus--;
  if(arus<0){arus=1000;}
  delay_ms(100);
 }
}
else if(display==2)
{lcd_gotoxy(1,0);
 lcd_putsf("SET          TIMER
BEBAN");
 ftoa(jeda,2,tempat);
 lcd_gotoxy(3,1);
 lcd_puts(tempat);
 lcd_gotoxy(0,1);
 lcd_putsf("PF=");
 if(Tplus==0)
 {jeda++;
  if(jeda>=59){jeda=0;}
  delay_ms(100);
 }
else if(Tmin==0)
 {jeda--;
  if(jeda<0){jeda=59;}
  delay_ms(100);
 }
}
if(display==4)
{freq_pln();
 tampilkan_arus();
 phase_beban();
 cosq_beban();
}

if(display==3)
{lcd_gotoxy(1,0);
 lcd_putsf("SET TIMER CEK");
 ftoa(cek,2,tempat);
 lcd_gotoxy(3,1);
 lcd_puts(tempat);
 lcd_gotoxy(0,1);
 lcd_putsf("PF=");
 if(Tplus==0)
 {cek++;
  if(cek>=59){cek=0;}
  delay_ms(100);
 }
else if(Tmin==0)
 {cek--;
  if(cek<0){cek=59;}
  delay_ms(100);
 }
}

/*
else if(display==3)
{lcd_gotoxy(1,0);
 lcd_putsf("SET COSPhi");
 ftoa(phase,2,tempat);
 lcd_gotoxy(3,1);
 lcd_puts(tempat);
 lcd_gotoxy(0,1);
 lcd_putsf("PF=");
 if(Tplus==0)
 {phase++;
  if(phase==256){phase=0;}
  delay_ms(10);
 }
else if(Tmin==0){phase--
;delay_ms(10);}
}
*/
if(display==0)
{delay_ms(50);
 lcd_clear();
}
else if(display>=1)
{delay_ms(50);
 lcd_clear();
}
}

```

Pada saat *variabel display* = 0 maka LCD akan menampilkan tampilan utama pembacaan tegangan, arus, beda fasa, watt, dan fungsi alat.

Pada saat *variabel display* = 1 maka lcd akan menampilkan tampilan untuk pengaturan batas arus.

Pada saat *variabel display* = 2 maka lcd akan menampilkan tampilan untuk pengaturan jeda waktu proteksi alat.

Pada saat *variabel display* = 3 maka lat akan menampilkan tampilan pengaturan jeda waktu pengecekan beban maksimal sebanyak tiga kali.

(3) *Sub rutin* pengolahan sampling beban

```
void olah_adc()
{if(read_adc(1)>0)
{temp=read_adc(1);
  if(temp1>temp    &&
  array_arus>=0    &&
  array_arus<=10   &&
  tanda_arus==0)

  {arus_adc[array_arus]=
  temp1;

  temp2=arus_adc[array
  _arus];

  array_arus=array_arus
  +1;
  }
  temp1=temp;timer_aru
  s=1;sec=0;
  }
  if(array_arus>10    &&
  tanda_arus==0)
  {for(i=0;i<=array_arus;i
  ++)}

  {if(temp3<arus_adc[i]){
  temp3=arus_adc[i];}
  }
  tanda_arus=1;
  }

  if(tanda_arus==1)
  {for(i=0;i<=array_arus;i
  ++)}

  {if(arus_adc[i]>0){arus
  _adc[i]=0;}}
  temp3=((temp3+temp2
  )/2);
  array_arus=0;tanda_ar
  us=0;
  }
  if(read_adc(1)==0    &&
  sec>2)

  {temp=0;temp1=0;tem
  p2=0;tanda_arus=0;te
  mp3=0;

  for(i=0;i<=array_arus;i
  ++)}

  {if(arus_adc[i]>0){arus
  _adc[i]=0;}}
  }

  timer_arus=0;array_ar
  us=0;sec=0;
  }
  //temp2=temp1;
  }
```

Pada program ini digunakan untuk proses penyamplingan rata-rata data tertinggi hasil pembacaan adc sensor arus sebanyak sepuluh kali

(4) Sub rutin konversi data

```
void      konversi(float
data_beban,      float
data_freq)
{if(data_beban>0)
{if(data_beban<=data_freq)
{a = ((data_freq -
data_beban)/data_freq);

if(data_beban<data_freq){b
= ((data_freq - data_beban)
* 10);}
c = b/data_freq;
d = (b - (data_freq * c));
if(d<data_freq){d = d *
10;}
e = d/data_freq;

f = (d - (data_freq * e));
if(f<data_freq){f = f * 10;}
g = f/data_beban;
hsl_cosp_i= a + (((c * 0.1) +
(e * 0.01)) + (g * 0.001));
}
}
if(data==0){hsl_cosp_i=freq/f
req;}
if(geser_phase==2      ||
geser_phase==0      &&
factor==0){and1=1;and2=0;}
if(geser_phase==3      &&
factor==2){and1=0;and2=1;}
}
```

Program ini digunakan untuk mengubah data hasil pengolahan beda fasa yang masih dalam bentuk data desimal ke dalam bentuk data desimal pecahan agar dapat ditampilkan ke LCD menjadi nilai pecahan.

(5) Sub rutin data prosesi

```
void data_prosesing()
{if(beban==1      &&
arus2>=arus3)
{arus_tanda=1;
//
if(sec1==0){relay=1;relay
=1;}
if(sec1>=jeda + 2)
{relay=1;
relay1=1;
beban=0;

opsi=opsi+1;
buzzer=1;
delay_ms(50);
opsi_waktu=1;

u_timer_opsi=0;timer_o
psi=0;
if(opsi>=1      &&
opsi<=2){buzzer=0;}
}
}
```

```

else if(arus2<arus3)
{sec1=0;
  arus_tanda=0;
}
if(beban==0 && opsi>=1
&& opsi<=2 &&
timer_opsi>cek){relay=0;
relay1=0;beban=1;sec1=
0;arus_tanda=0;opsi_wa
ktu=0;u_timer_opsi=0;ti
mer_opsi=0;delay_ms(50
);}
if(beban==0 &&
opsi>2){relay=1;relay1=1
;beban=0;opsi_waktu=0;

opsi=0;u_timer_opsi=0;s
ec1=0;arus_tanda=0;tim
er_opsi=0;delay_ms(200)
;buzzer=0;}
if(Tplus==0 &&
beban==0 && display==0
&& relay==1 &&
relay1==1)
{relay=0;relay1=0;beban
=1;opsi=0;opsi_waktu=0;
u_timer_opsi=0;sec1=0;a
rus_tanda=0;timer_opsi=
0;delay_ms(50);} //aktif
kan relay
}

```

Program ini digunakan proses pengolahan data pembacaan semua sensor untuk proses proteksi pemutusan arus ke beban dan mebunyikan *buzzer*.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari uraian perancangan, pembuatan dan pembahasan tentang Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus maka dapat disimpulkan :

1. Perangkat keras (*Hardware*) Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus terdiri dari rangkaian *power supply*, rangkaian sensor tegangan , sensor arus *Split Current transformer CT301*, rangkaian penguat instrumentasi, rangkaian sistem minimum Amega 16, rangkaian pendeteksi beda fasa sebagai pembaca *cos phi*, rangkaian driver *relay* dan *buzzer* sebagai pemutus arus saat beban berlebih dan *buzzer* berfungsi sebagai alarm.
2. Perangkat lunak Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus berupa program bahasa C dibuat menggunakan kompiler *Code Vision AVR* , yang terdiri dari beberapa bagian: Definisi prosesor, Penyertaan fungsi, Definisi *Port*, *Mode ADC*, *Mode Interrupt* dan *Mode Timer*, Deklarasi variabel serta Fungsi Utama kemudian di *download*kan ke mikrokontroler ATmega 16 dengan *Downloder*.
3. Dari pengujian yang dilakukan Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus secara keseluruhan sudah berfungsi namun masih ada beberapa kekurangan kurang presisinya hasil

pembacaan pengukuran nilai arus (A), nilai Factor Daya (*cosphi*), dan nilai daya (W)

B. Keterbatasan Alat

Alat Sistem Proteksi Daya Listrik Berbasis Mikrokontroler ATmega 16 dengan Sensor Arus mempunyai keterbatasan:

1. Hanya dapat mengukur arus maksimal 2 A
2. Hasil pembacaan nilai Arus, *Cosphi*, dan Daya masih kurang akurat dibandingkan dengan alat ukur pembanding.
3. Pembacaan pergeseran beda fasa secara simulasi berbeda dengan kenyataan disebabkan oleh perbedaan bahan jenis antara sensor tegangan (trafo) yang berbahan besi lunak sedangkan sensor arus berbahan jenis Ferit sehingga saat dilakukan pengukuran dengan CRO sudah terjadi pergeseran beda fasa pada saat beban resistif dimana karakteristik fasa dari ferit mendahului bahan dari besi lunak.
4. Dengan terdapatnya perbedaan pembacaan nilai Arus dan nilai *Cosphi* maka nilai Daya juga akan berbeda.
5. Rangkaian terkadang *restart* sendiri disebabkan ketidak stabilan komponen mikrokontroler ATmega 16 pada posisi beban Induktif.

C. Saran

Dalam pembuatan proyek akhir ini terdapat kekurangan seperti yang disebutkan dalam point keterbatasan alat, sehingga diperlukan pengembangan

guna menyempurnakan proyek akhir ini. Penulis mempunyai beberapa saran untuk menyempurnakan alat ini dari keterbatasannya diantaranya :

1. Menambahkan rangkaian *range selector* sehingga dapat dapat mengukur beban lebih dari 2 A .
2. Mengganti sensor arus dengan bahan jenis yang sama dengan sensor tegangan sehingga pada saat beban resistif tidak terjadi pergeseran beda fasa..
3. Mengganti komponen dengan komponen kelas industri sehingga kinerja alat bisa lebih stabil.

DAFTAR PUSTAKA

- Andrianto Heri. (2008). *Pemrograman Mikrokontroler ATmega16 Menggunakan Code Vision AVR*. Bandung: Informatika.
- ATMEL Corp. (2009). *Datasheet ATmega16(L)*, Diakses pada tanggal 22 Agustus 2014, dari www.atmel.com/images/doc2466.pdf
- Pree Kang. (2014). *Trafo atau Transformator* Diakses pada 11 Februari 2014 dari <http://www.elektronika123.com/trafo-atau-transformator/>
- Elektronika-dasar.web.id. (2012). *Teori Relay* Diakses pada tanggal 11 Februari 2014 dari <http://elektronika-dasar.web.id/teori-elektronika/teori-relay-elektro-mekanik/>
- Herlambang. (2010). *Pengertian Sensor*. Diakses pada tanggal 22 Agustus 2014, dari <http://www.nubielab.com/elektronika/analog/pengertian-sensor>
- Heriyanto Ary M. dan Adi Wisnu P (2008). *Pemograman Bahasa C untuk Mikrokontroler ATmega8535*. Yogyakarta: Andi
- Indraharja. (2012). *Pengertian Buzzer*. Diakses tanggal 20 Agustus 2014, dari <http://indraharja.wordpress.com/2012/01/07/pengertian-buzzer/>
- Kho Dickson. (2014). *Pengertian Gerbang Logika Dasar dan jenis-jenisnya*. Diakses pada tanggal 25 Desember 2014 dari <http://teknikelektronika.com/pengertian-gerbang-logika-dasar-simbol/>
- Rizal Syamsul. (2011). *Liquid Crystal Diode (LCD)*. Diakses tanggal 20 Agustus 2014, dari <http://id.scribd.com/doc/50403145/Liquid-Crystal-Diode-LCD>
- Setiawan Iwan. (2006). *Tutorial Mikrokontroler AVR*. Diakses tanggal 20

Agustus 2014 dari <http://id.scribd.com/doc/181041481/Tutorial-Microcontroller-AVR-Part-I-pdf>

Sudira Putu. (2006). *Algoritma Pemrograman Mikroprosesor*. Departemen Pendidikan Nasional Fakultas Teknik Universitas Negeri Yogyakarta

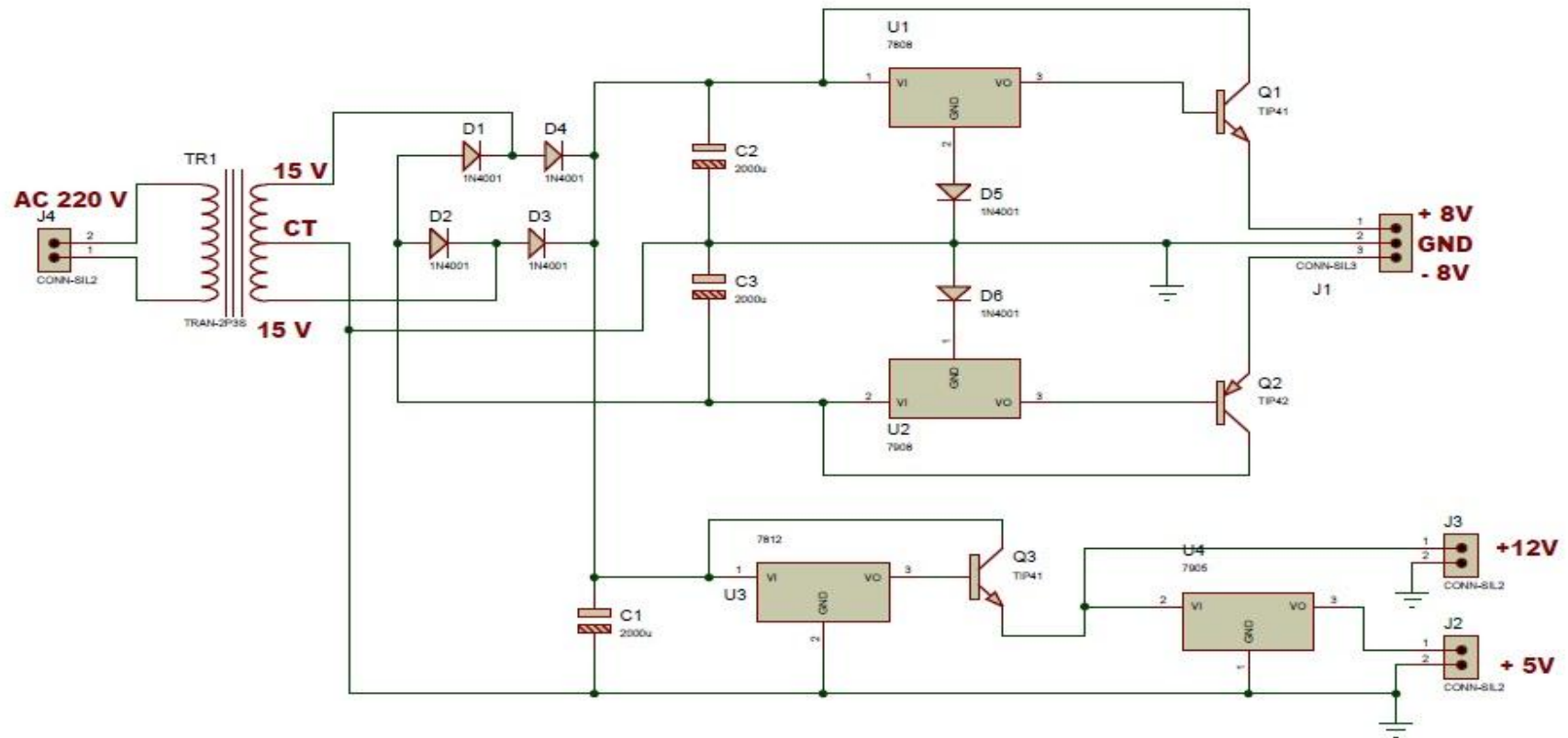
TEXAS INSTRUMENTS Corp. (2010). *Datasheet LM358* Diakses pada tanggal 22 Agustus 2014, dari www.ti.com/lit/ds/symlink/lm358.pdf

TEXAS INSTRUMENTS Corp. (2010). *Datasheet INA122* Diakses pada tanggal 22 Agustus 2014, dari www.ti.com/lit/ds/symlink/ina122.pdf

WIKIPEDIA.(2014) *Pengertian Sensor* Diakses pada tanggal 11 Februari 2014,dari <http://id.wikipedia.org/wiki/Sensor>

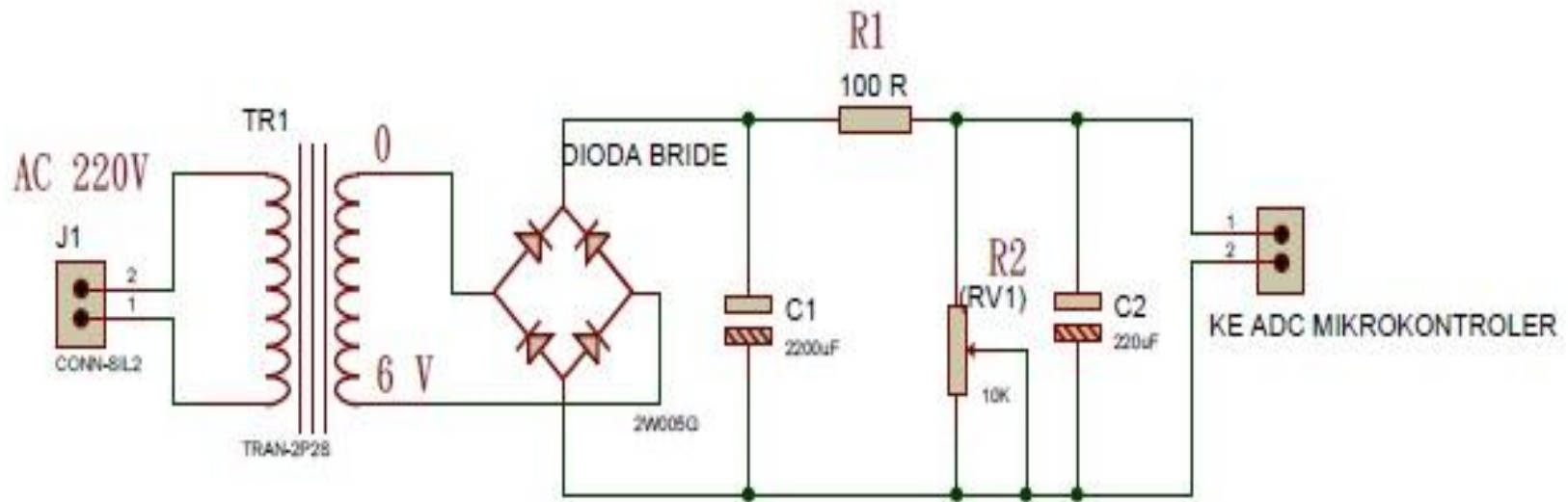
LAMPIRAN

Lampiran 1. Skema Rangkaian *Power Supply*



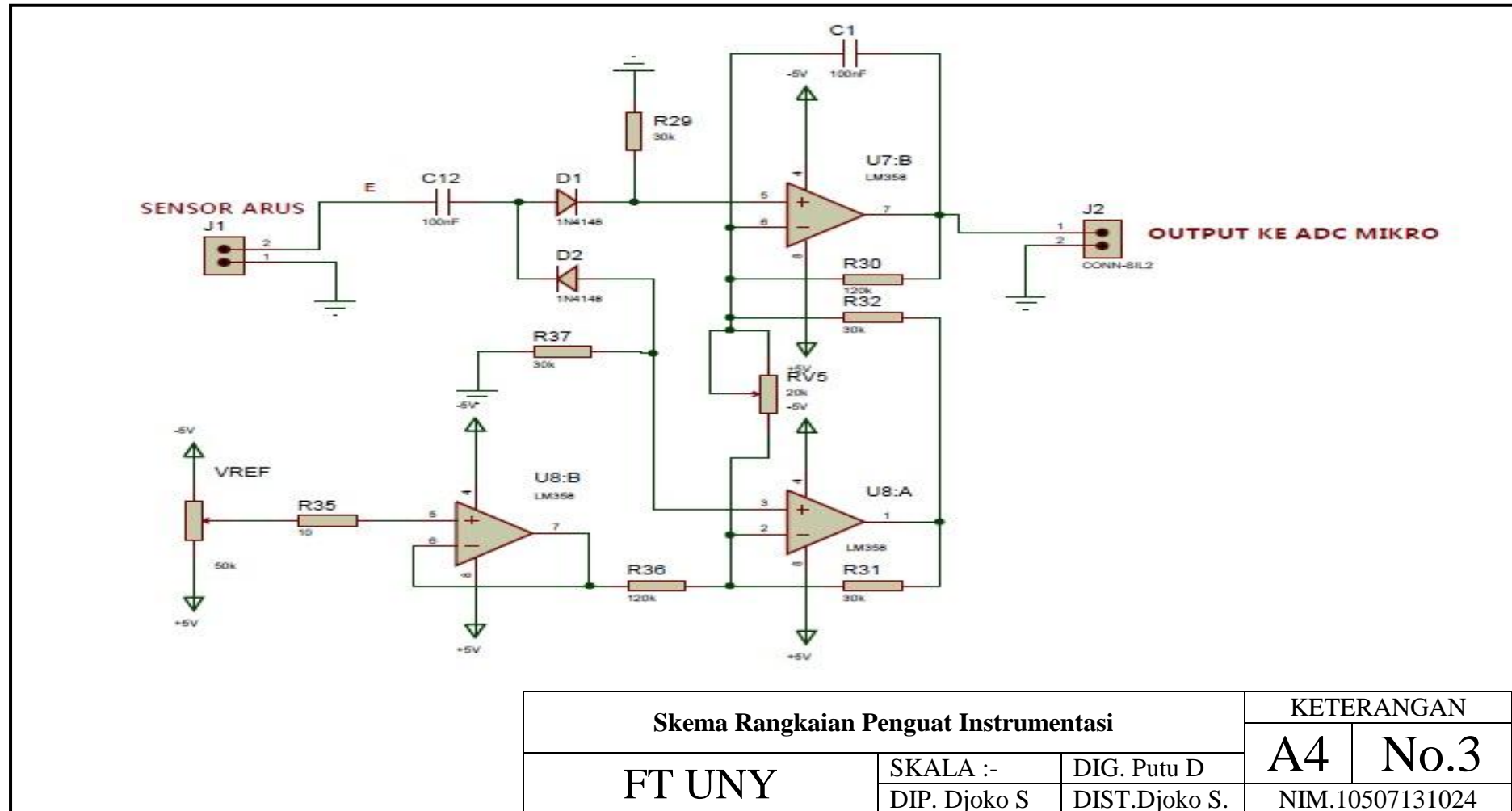
Skema Rangkaian <i>Power Supply</i>			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.1
	DIP. Djoko S	DIST.Djoko S	NIM.10507131024	

Lampiran 2. Skema Rangkaian Sensor Tegangan

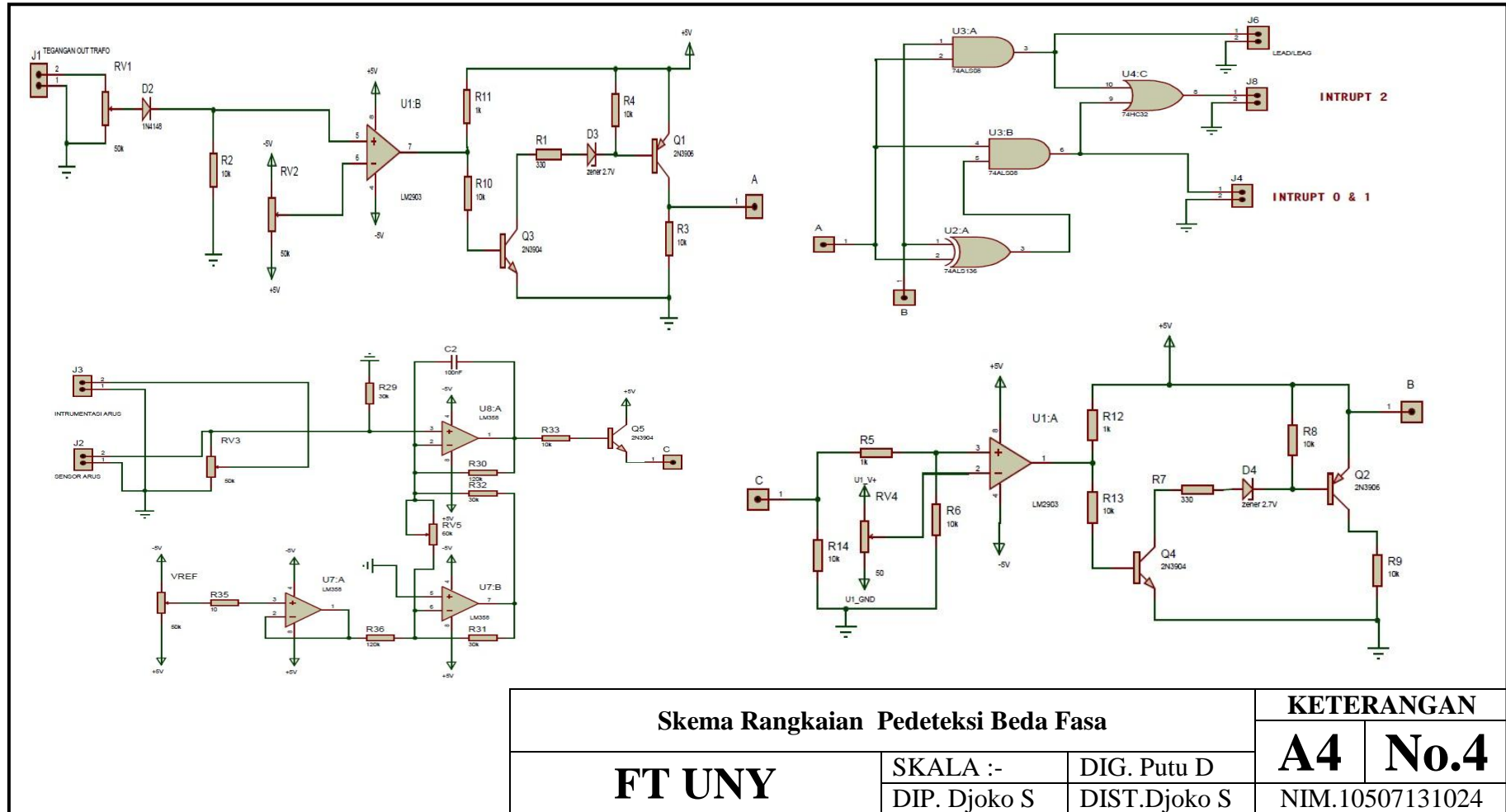


Skema Rangkaian Sensor Tegangan			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.2
	DIP. Djoko S	DIST.Djoko S	NIM.10507131024	

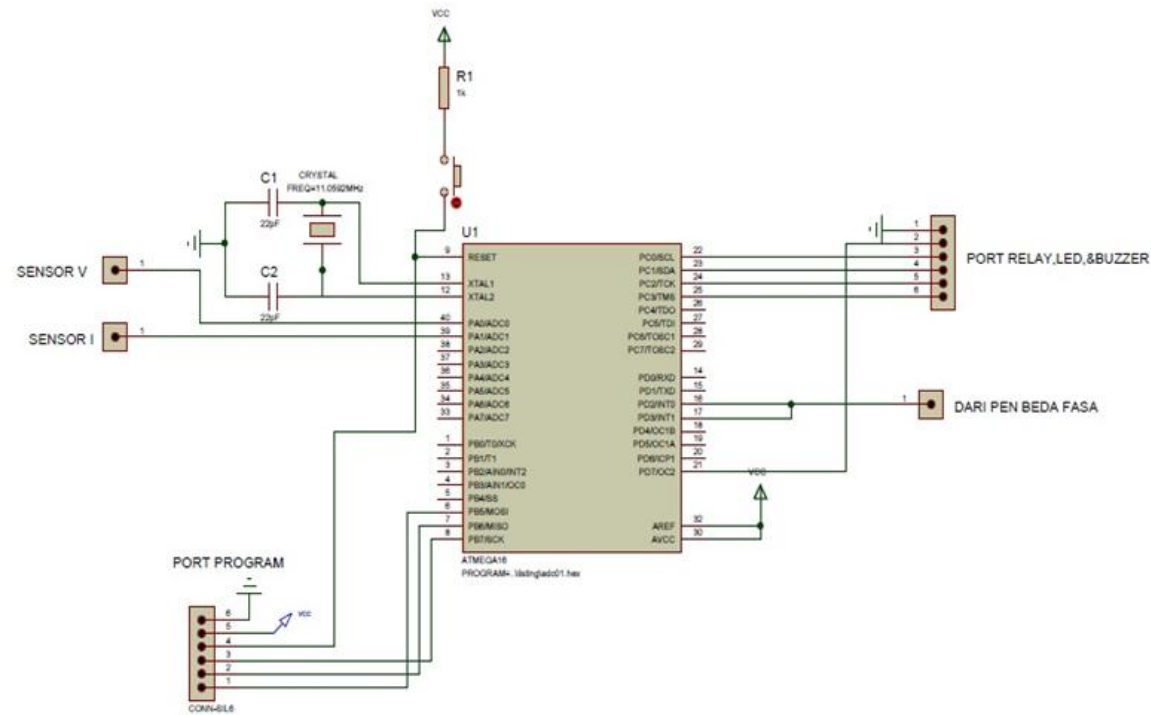
Lampiran 3. Skema Rangkaian Penguat Instrumentasi



Lampiran 4. Skema Rangkaian Pendeteksi Beda Fasa

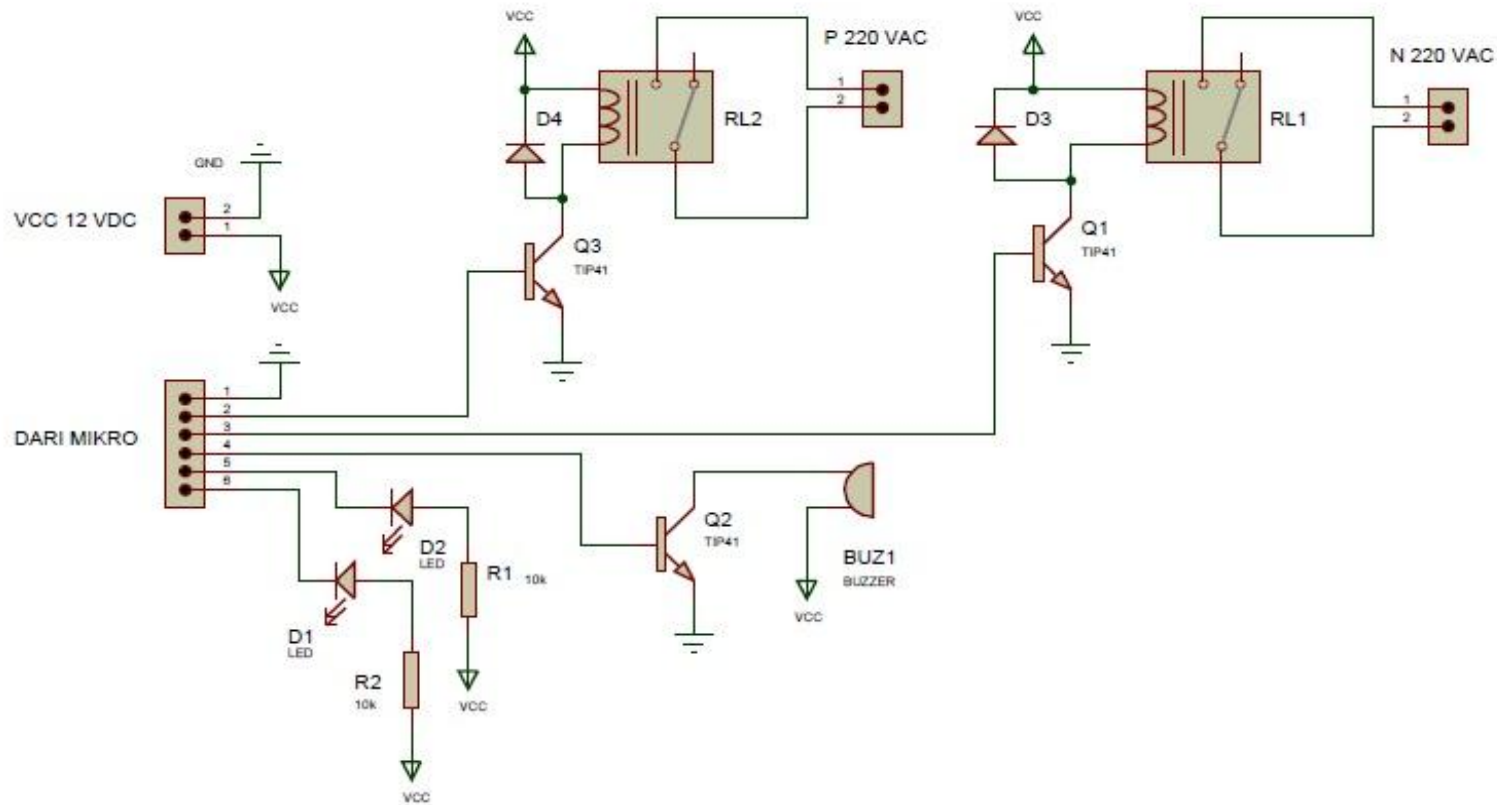


Lampiran 5. Skema Rangkaian Sistem Minimum Mikrokontroler Atmega 16



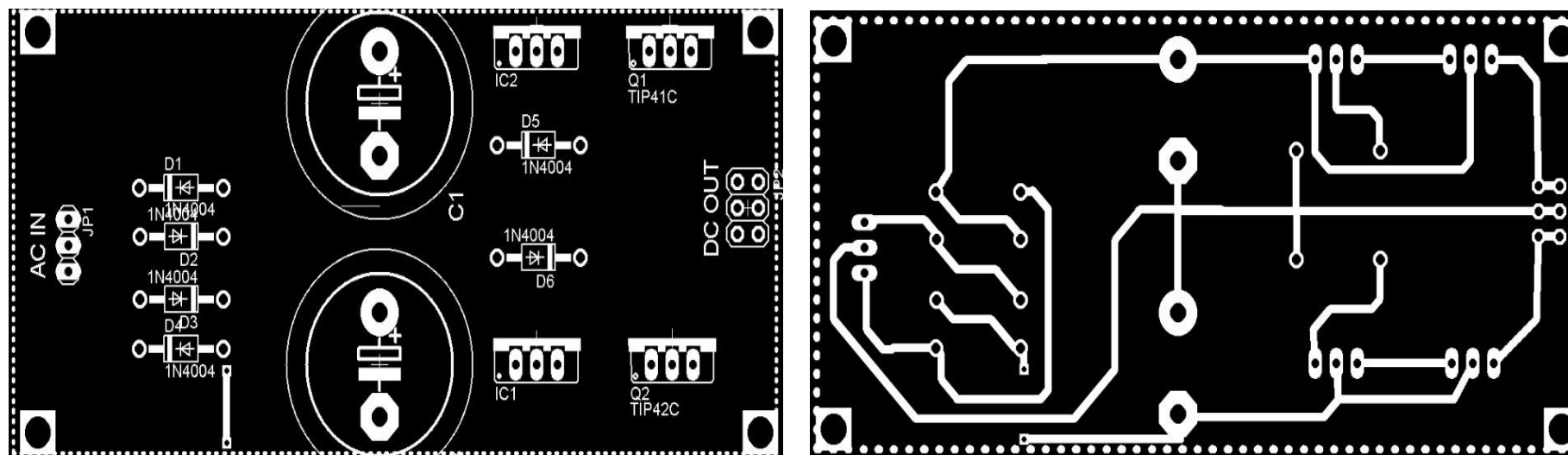
Skema Rangkaian Sistem Minimum Mikrokontroler ATmega 16			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.5
	DIP. Djoko S	DIST. Djoko S	NIM.09507134010	

Lampiran 6. Skema Rangkaian *Driver Relay* dan *Buzzer*



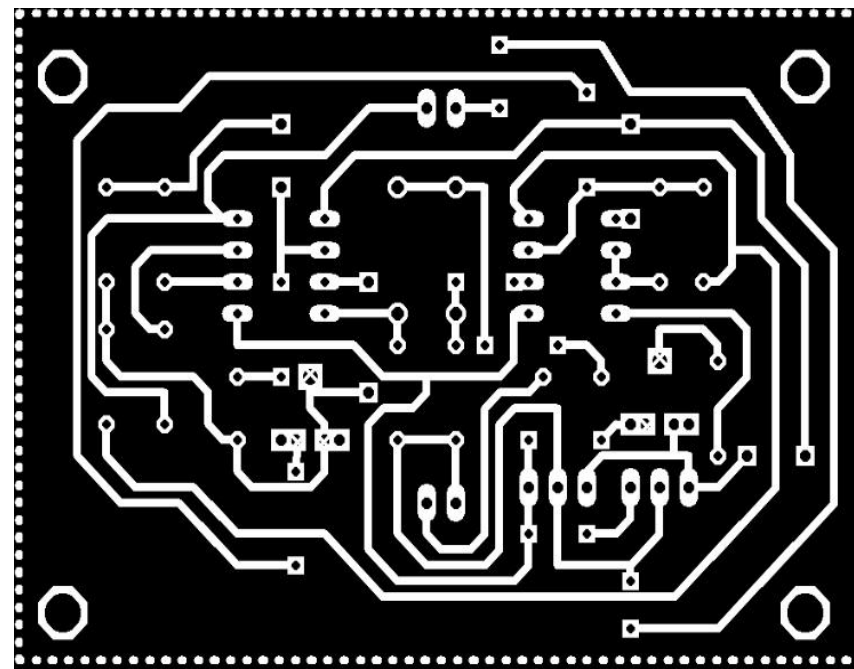
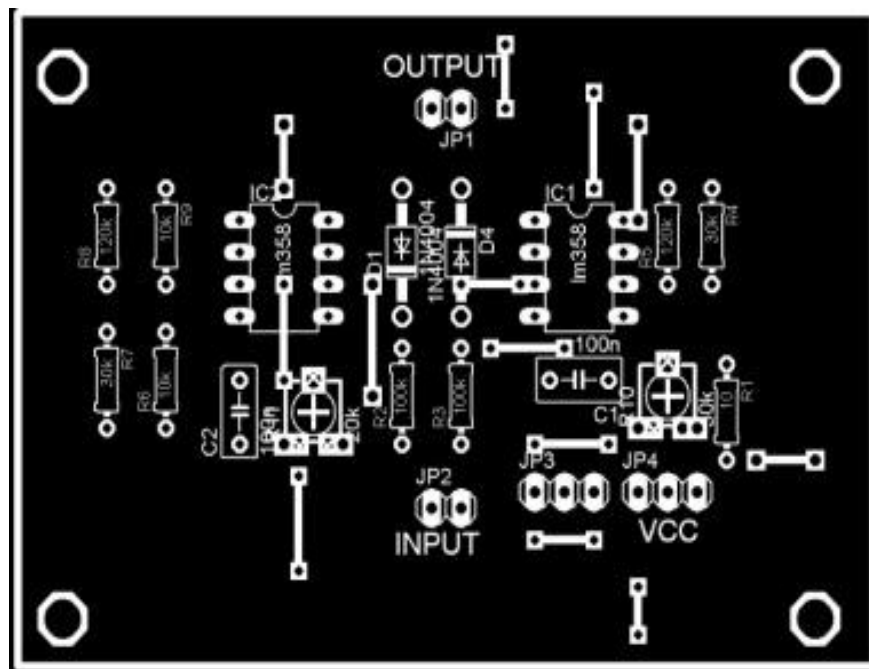
Skema Rangkaian <i>Driver Relay dan Buzzer</i>			KETERANGAN	
			A4	No.6
FT UNY	SKALA :-	DIG. Putu D	NIM.10507131024	
	DIP. Djoko S	DIST. Djoko S		

Lampiran 7. Tata letak komponen dan *layout* PCB Rangkaian *Power Supply*



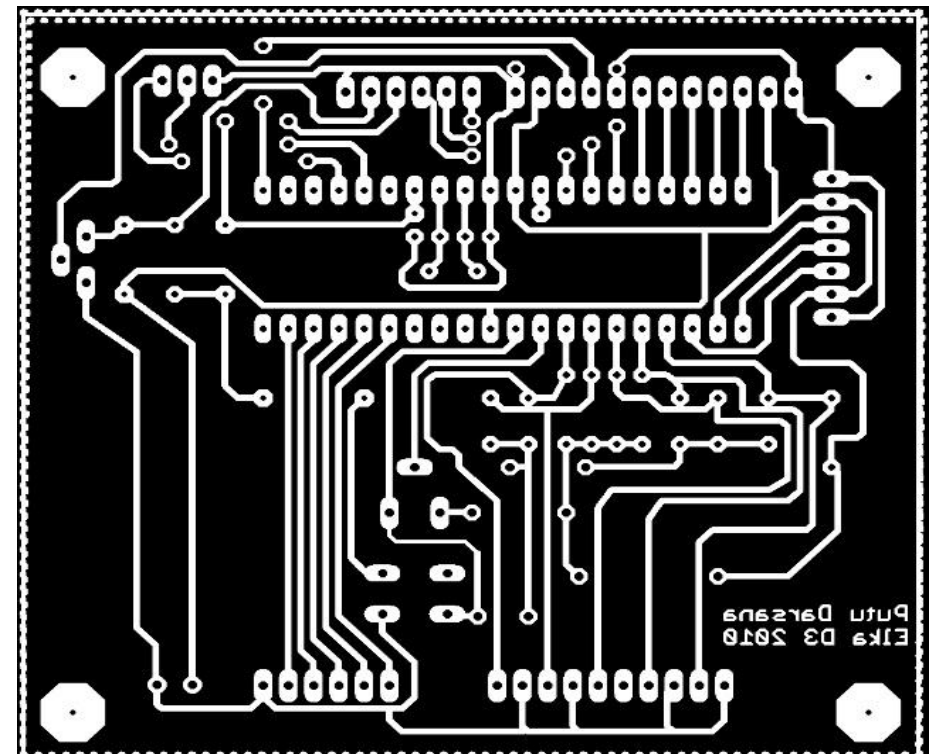
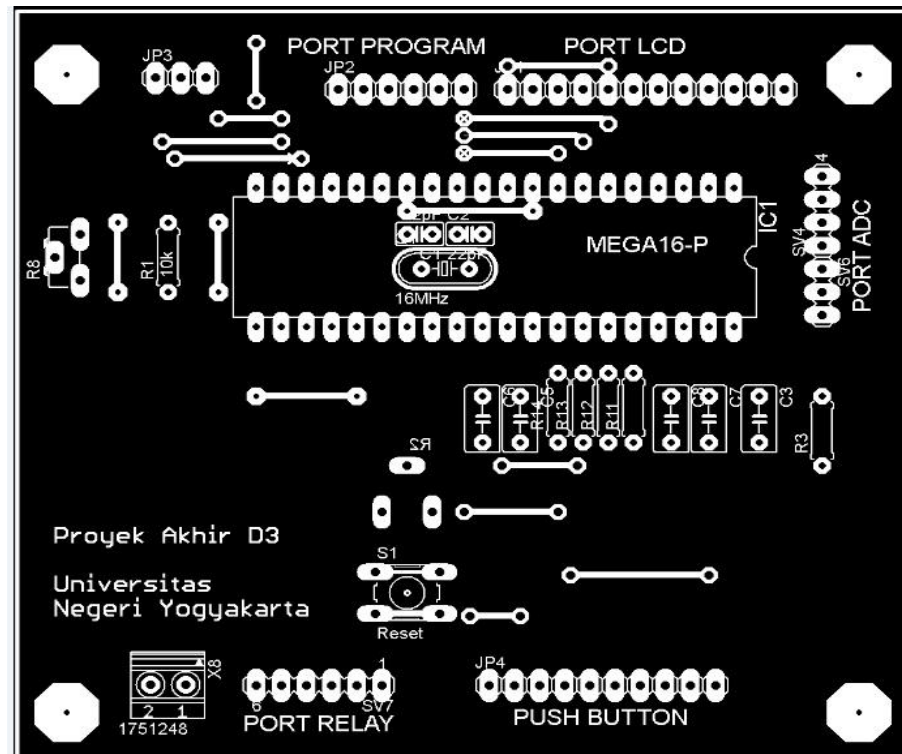
Tata letak komponen dan layout PCB rangkaian <i>Power Supply</i>			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.7
	DIP. Djoko S	DIST. Djoko S	NIM.10507131024	

Lampiran 8. Tata letak komponen dan *layout* PCB Rangkaian Penguat Instrumentasi



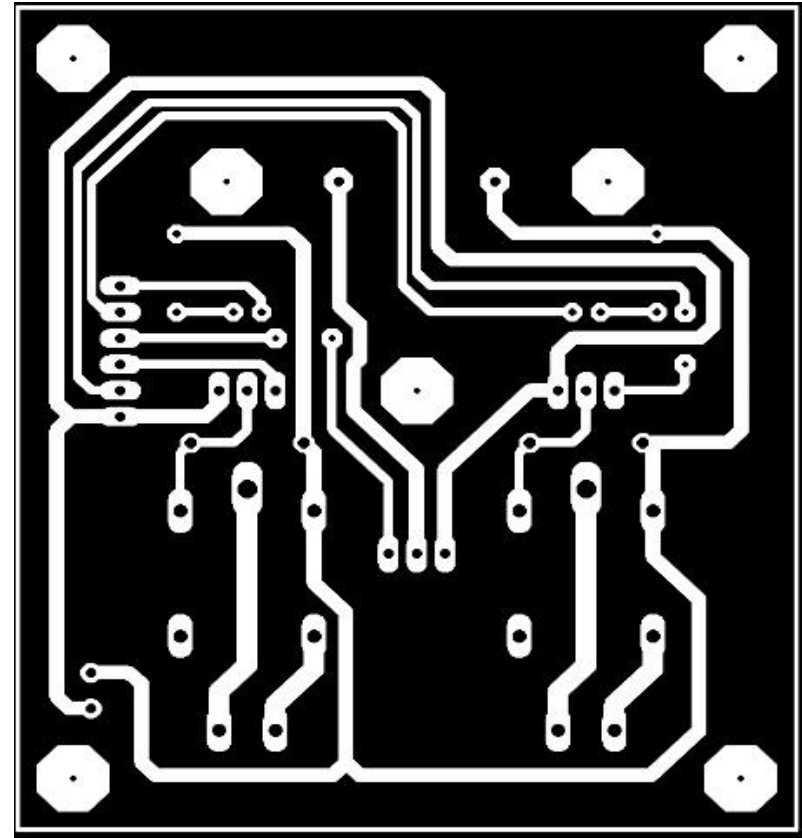
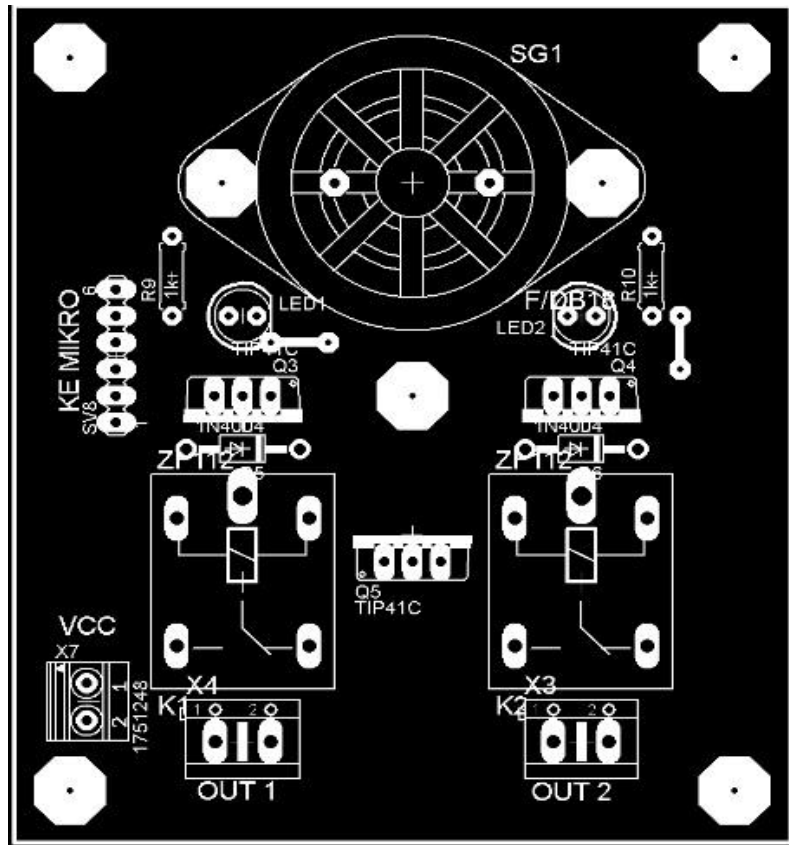
Tata letak komponen dan layout PCB rangkaian Penguat Instrumentasi			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.8
	DIP. Djoko S	DIST. Djoko S	NIM.10507131024	

Lampiran 9. Tata letak komponen dan *layout* PCB Rangkaian Sistem Minimum Mikrokontroler Atmega 16



Tata letak komponen dan layout PCB rangkaian Sistem Minimum Mikrokontroler Atmega 16			KETERANGAN	
FT UNY	SKALA :-	DIG. Putu D	A4	No.9
	DIP. Djoko S	DIST. Djoko S	NIM.10507131024	

Lampiran 10. Tata letak komponen dan *layout* PCB Rangkaian *Driver relay* dan *Buzzer*



Tata letak komponen dan layout PCB rangkaian *Driver Relay* dan *Buzzer*

KETERANGAN

A4 **No.10**

FT UNY

SKALA :-

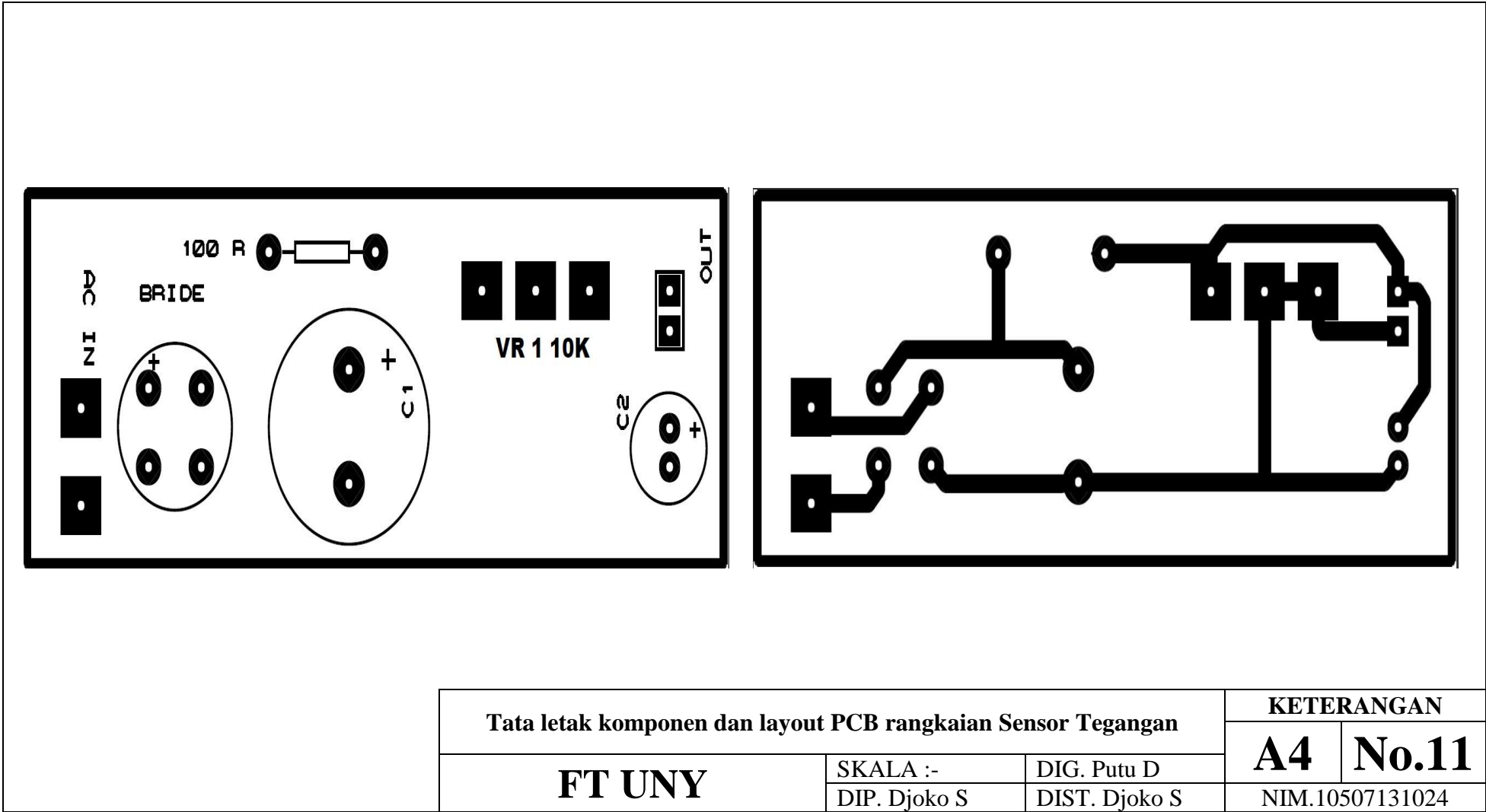
DIP. Djoko S

DIG. Putu D

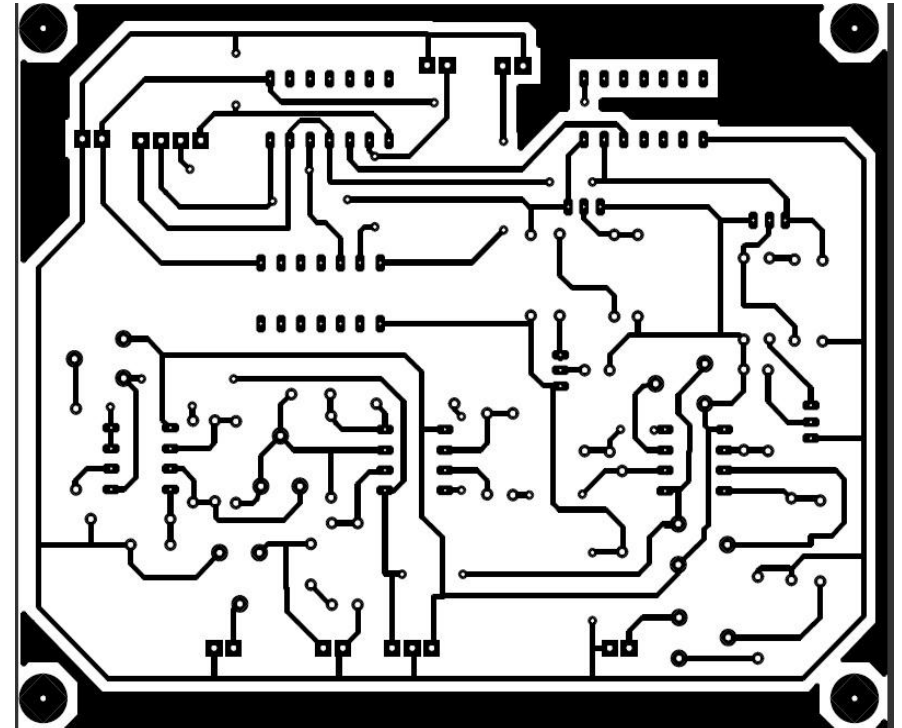
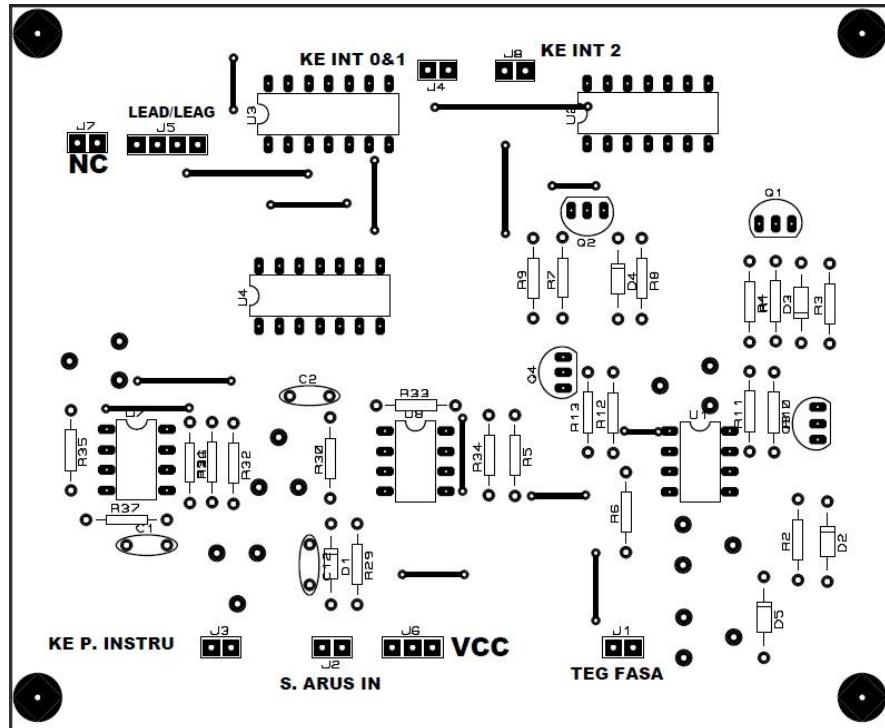
DIST. Djoko S

NIM.10507131024

Lampiran 11. Tata letak komponen dan *layout* PCB Rangkaian Sensor Tegangan



Lampiran 12. Tata letak komponen dan *layout* PCB Rangkaian Pendeteksi Beda Fasa



Tata letak komponen dan layout PCB rangkaian Pendeteksi Beda Fasa

KETERANGAN

A4 **No.12**

FT UNY

SKALA :-

DIP. Djoko S

DIG. Putu D

DIST. Djoko S

NIM.10507131024

Lampiran 13. Program Lengkap

/******

This program was produced by the

CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 01/10/2014

Author : PerTic@n

Company : If You Like This Software,Buy It

Comments:

Chip type : ATmega16A

Program type : Application

AVR Core Clock frequency: 12,000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

#include <mega16a.h>

#include <stdlib.h>

#include <delay.h>

// Alphanumeric LCD functions

#include <alcd.h>

float vin,iin,watt,data,freq,hsl_cospi,SP,arus1;

int tanda,tanda1,cv,display=0,timer_arus=0,timer_volt=0,arus_adc[20];

unsigned int temp,temp1=0,temp2,arus2=0,arus3=0,array_arus=0;

long int usec,sec,usec1,sec1;

unsigned long int a,b,c,d,e,f,g;

char tempat[33],arus_tanda=0,beban=0,geser_phase=0,factor=0;

eprom signed int arus,jeda,cek;

int opsi=0;

int i,k,temp3=0,tanda_arus=0,tanda_buzzer=0,clear_lcd=0;

#define menu1 PINA.7

#define lplus PINA.5

#define lmin PINA.6

#define Tplus PINA.3

#define Tmin PINA.4

#define buzzer PORTC.2

#define relay PORTC.1

#define relay1 PORTC.0

#define and1 PORTC.3

#define and2 PORTC.4

#define corect_v PIND.6

#define corect_i PIND.5

// External Interrupt 0 service routine

interrupt [EXT_INT0] void ext_int0_isr(void)

```

{
// Place your code here
if(corect_v==1 && arus2>0)
{tanda=1;
} //sinyal arus

}
/*
if(data>0)
{if(geser_phase==2){lcd_putsf("-");}
if(geser_phase==3){lcd_putsf("+");}
}

*/
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
if(arus2>0 && factor==0){geser_phase=2;tanda=0;}
if(arus2>0 && factor==2){geser_phase=3;tanda=0;}
//if(data>freq && PIND.1==1 && arus2>0){factor=2;}
//if(data>freq && PIND.1==0 && arus2>0){factor=0;}
}

interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
if(corect_v==1 && tanda1==0){tanda1=1;}

```

```

if(corect_v==1 && corect_i==0){factor=0;}
if(corect_v==1 && corect_i==1){factor=2;}

}

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x6A;
// Place your code here
if(++usec1==10000)
{if(++sec1==60 && arus_tanda==1){sec1=0;}
usec1=0;
}
}
int secv,opsi_waktu=0,timer_opsi=0,u_timer_opsi=0;
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
TCNT1H=0xFF;
TCNT1L=0x6A;
// Place your code here
if(tanda==1 && arus2>0)
{data=data+1;
}
if(++usec==10000)
{if(++sec==60 && timer_arus==1){sec=0;}

```

```

    if(++secv==6 && timer_volt==1){secv=0;}
    usec=0;
}
if(tanda1==1)
{freq=freq+1;
}
if(opsi_waktu==1)
{if(++u_timer_opsi==10000)
{if(++timer_opsi==60){timer_opsi=0;}
u_timer_opsi=0;
}
}
if(++i==1000 && tanda_buzzer==1)
{if(++k==10){k=0;}
buzzer=k%2;
i=0;
}
}

#define ADC_VREF_TYPE 0x00
// #define ADC_VREF_TYPE 0x40
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion

```

```

    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

void conversi(float data_beban, float data_freq)
{if(data_beban>0)
{if(data_beban<=data_freq)
{a = ((data_freq - data_beban)/data_freq);
if(data_beban<data_freq){b = ((data_freq - data_beban) * 10);}
c = b/data_freq;
d = (b - (data_freq * c));
if(d<data_freq){d = d * 10;}
e = d/data_freq;
f = (d - (data_freq * e));
if(f<data_freq){f = f * 10;}
g = f/data_beban;
hsl_cospi= a + (((c * 0.1) + (e * 0.01)) + (g * 0.001));

}
}
if(data==0){hsl_cospi=freq/freq;}
if(geser_phase==2 || geser_phase==0 && factor==0){and1=1;and2=0;}
if(geser_phase==3 && factor==2){and1=0;and2=1;}
}

```

```

void koreksi_signalv()
{if(corect_v==0 && tanda1==1){tanda1=0;}
 if(tanda1==1 && arus2<=0){tanda=0;}
}

```

```

void freq_pln()
{ftoa(freq,0,tempat);
 lcd_gotoxy(2,0);
 lcd_puts(tempat);
 lcd_gotoxy(0,0);
 lcd_putsf("T=");
}

```

```

void phase_beban()
{ftoa(data,0,tempat);
 lcd_gotoxy(2,1);
 lcd_puts(tempat);
 lcd_gotoxy(0,1);
 lcd_putsf("P=");
 conversi(data,freq);
}

```

```

void cosq_beban()
{ftoa(hsl_cospi,3,tempat);
 lcd_gotoxy(11,1);
 lcd_puts(tempat);
 lcd_gotoxy(10,1);
 if(data>0)

```

```

    {if(geser_phase==2){lcd_putsf("+");}
      if(geser_phase==3){lcd_putsf("-");}
    }
    // if(data==0){lcd_putsf(" ");}

```

```

    delay_ms(1);
    freq=0;
    data=0;
    conversi(0,0);

}

```

```

void tampilkan_tegangan()
{ temp=read_adc(0);
 // vin=(float)temp*0.00488758*44; //tegangan max 220 Volt
 vin=(float)temp*0.0055*44;
 // ftoa(arus2,1,tempat);
 ftoa(vin,1,tempat);
 lcd_gotoxy(0,0);
 lcd_putsf("V=");
 lcd_gotoxy(2,0);
 lcd_puts(tempat);
 lcd_gotoxy(7,0);
 lcd_putsf("V");
}

```

```

void olah_adc()
{if(read_adc(1)>0)
{temp=read_adc(1);

```

```

if(temp1>temp && array_arus>=0 && array_arus<=10 &&
tanda_arus==0)
{
    arus_adc[array_arus]=temp1;
    temp2=arus_adc[array_arus];
    array_arus=array_arus+1;
}
temp1=temp;timer_arus=1;sec=0;
}
if(array_arus>10 && tanda_arus==0)
{for(i=0;i<=array_arus;i++)
    {if(temp3<arus_adc[i]){temp3=arus_adc[i];}
    }
tanda_arus=1;
}
if(tanda_arus==1)
{for(i=0;i<=array_arus;i++)
    {if(arus_adc[i]>0){arus_adc[i]=0;}}
temp3=((temp3+temp2)/2);
array_arus=0;tanda_arus=0;
}
if(read_adc(1)==0 && sec>2)
{temp=0;temp1=0;temp2=0;tanda_arus=0;temp3=0;
for(i=0;i<=array_arus;i++)
    {if(arus_adc[i]>0){arus_adc[i]=0;}
    }
timer_arus=0;array_arus=0;sec=0;
}
//temp2=temp1;

```

```

}
void tampilkan_arus()
{
    iin=(float)temp3*0.00196875; // batas ukur 2A
    arus2=iin * 100;
    //iin=temp;
    ftoa(iin,2,tempat);
    lcd_gotoxy(9,0);
    lcd_putsf("I=");
    lcd_gotoxy(11,0);
    lcd_puts(tempat);
    lcd_gotoxy(15,0);
    lcd_putsf("A");
}
void tampilkan_cos()
{
    //SP=0.0003846;
    //if(arus2>0){cospi=SP*data;}
    //if(arus2==0){cospi=0;}
    ftoa(hsl_cospi,3,tempat);
    lcd_gotoxy(1,1);
    lcd_puts(tempat);
    lcd_gotoxy(0,1);
    if(data>0)
        {if(geser_phase==2){lcd_putsf("+");}
        if(geser_phase==3){lcd_putsf("-");}
        }
    // if(data==0){lcd_putsf(" ");}
    // delay_ms(1);
}

```

```

// data=0;

}
void tampilkan_watt()
{
    watt=vin*iin*hsl_cospi;
    ftoa(watt,2,tempat);
// ftoa(read_adc(1),2,tempat);
    lcd_gotoxy(9,1);
    lcd_puts(tempat);
    lcd_gotoxy(15,1);
    lcd_putsf("W");
    delay_ms(1);
    freq=0;
    data=0;
    conversi(0,0);
// temp3=0;
}
void menu()
{if(menu1==0)
    {display++;
    if(display==5){display=0;}
    delay_ms(100);
    }
if(display==0)
    {tampilkan_tegangan();
    arus1=(float)arus*0.01;
    arus3=arus1 * 100;

```

```

    tampilkan_arus();
    conversi(data,freq);
    tampilkan_cos();
    tampilkan_watt();}
else if(display==1)
    {lcd_gotoxy(1,0);
    lcd_putsf("SET ARUS");
    arus1=(float)arus*0.01;
    arus3=arus1 * 100;
    ftoa(arus1,2,tempat);
    lcd_gotoxy(3,1);
    lcd_puts(tempat);
    lcd_gotoxy(0,1);
    lcd_putsf("A =");
    if(lplus==0)
        {arus++;
        if(arus>=1000){arus=0;}
        delay_ms(100);
        }
    else if(lmin==0)
        {arus--;
        if(arus<0){arus=1000;}
        delay_ms(100);
        }
    }
else if(display==2)
    {lcd_gotoxy(1,0);

```



```

lcd_putsf("SET TIMER BEBAN");
ftoa(jeda,2,tempat);
lcd_gotoxy(3,1);
lcd_puts(tempat);
lcd_gotoxy(0,1);
lcd_putsf("TB=");
if(Tplus==0)
{
jeda++;
if(jeda>=59){jeda=0;}
delay_ms(100);
}
else if(Tmin==0)
{
jeda--;
if(jeda<0){jeda=59;}
delay_ms(100);
}
}

if(display==4)
{
freq_pln();
tampilkan_arus();
phase_beban();
cosq_beban();
}

if(display==3)
{
lcd_gotoxy(1,0);
lcd_putsf("SET TIMER CEK");

```

```

ftoa(cek,2,tempat);
lcd_gotoxy(3,1);
lcd_puts(tempat);
lcd_gotoxy(0,1);
lcd_putsf("TC=");
if(Tplus==0)
{
cek++;
if(cek>=59){cek=0;}
delay_ms(100);
}
else if(Tmin==0)
{
cek--;
if(cek<0){cek=59;}
delay_ms(100);
}
}

if(display==0)
{
delay_ms(50);
lcd_clear();
}
else if(display>=1)
{
delay_ms(50);
lcd_clear();
}
}

```

```

void data_procesing()
{if(beban==1 && arus2>=arus3)
{arus_tanda=1;
if(sec1>=jeda + 2)
{relay=1;relay1=1;beban=0;
opsi=opsi+1;
buzzer=1;delay_ms(50);
opsi_waktu=1;u_timer_opsi=0;timer_opsi=0;
if(opsi>=1 && opsi<=2){buzzer=0;}
}
}
else if(arus2<arus3){sec1=0;arus_tanda=0;}
if(beban==0 && opsi>=1 && opsi<=2 &&
timer_opsi>cek){relay=0;relay1=0;beban=1;sec1=0;arus_tanda=0;opsi_w
aktu=0;u_timer_opsi=0;timer_opsi=0;delay_ms(50);}
if(beban==0 &&
opsi>2){relay=1;relay1=1;beban=0;opsi_waktu=0;opsi=0;u_timer_opsi=0;
sec1=0;arus_tanda=0;timer_opsi=0;buzzer=0;tanda_buzzer=1;delay_ms(
50);}
if(Tplus==0 && beban==0 && display==0 && relay==1 && relay1==1)
{relay=0;relay1=0;beban=1;opsi=0;opsi_waktu=0;u_timer_opsi=0;sec1=0;
arus_tanda=0;timer_opsi=0;i=0;k=0;tanda_buzzer=0;buzzer=0;delay_ms(
50);}
//else if(Tplus==0 && beban==1 && relay==1 &&
relay1==1){relay=0;relay1=0;beban=0;delay_ms(50);}//aktif kan relay
//relay=0;
//relay1=0;
}

```

```

// Declare your global variables here

```

```

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0xF8;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T

```

```

PORTC=0x00;
DDRC=0x1F;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x02;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 12000,000 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge

```

```

// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x02;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge

```

```

// INT1: On
// INT1 Mode: Falling Edge
// INT2: Off
//GICR|=0xC0;
//MCUCR=0x0A;
//MCUCSR=0x00;
//GIFR=0xC0;

//GICR|=0xC0;
//MCUCR=0x0B;
//MCUCSR=0x00;
//GIFR=0xC0;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: On
// INT1 Mode: Falling Edge pot to neg signal
// INT2: On
// INT2 Mode: Rising Edge neg to pot signal
//GICR|=0xE0;
//MCUCR=0x0B;
//MCUCSR=0x00;
//GIFR=0xE0;
GICR|=0xE0;
MCUCR=0x0B;
MCUCSR=0x40;

```

```

GIFR=0xE0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x05;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 750,000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: Free Running
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

//ADMUX=ADC_VREF_TYPE & 0xff;
//ADCSRA=0xA4;
//SFIOR&=0x1F;
//ADMUX=ADC_VREF_TYPE & 0xff;
//ADCSRA=0xA4;

```

```

//SFIOR&=0x1F;
//SFIOR|=0xC0;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

lcd_init(16);

// Global enable interrupts
#asm("sei")
lcd_gotoxy(1,0);
lcd_putsf("TUGAS AKHIR");
delay_ms(10);

lcd_gotoxy(2,1);
lcd_putsf("PHUTU DARSANA");
delay_ms(1000);
lcd_clear();

lcd_gotoxy(0,0);
lcd_putsf("Cosphi Watt Meter ATmega16");
delay_ms(1000);
lcd_clear();

```

```

cv=1;
SP=0.0010769230;
relay=1;
relay1=1;
buzzer=1;
delay_ms(50);
buzzer=0;

//SP=0.000885;
while (1)
    {
        //adc_volt();
        koreksi_signalv();
        olah_adc();
        data_procesing();
        menu();
        // Place your code here
    }
}

```